



**TURUN  
YLIOPISTO**  
UNIVERSITY  
OF TURKU

# UNDERSTANDING THE STRUCTURE AND MEANING OF FINNISH TEXTS: FROM CORPUS CREATION TO DEEP LANGUAGE MODELLING

---

Jenna Kanerva





**TURUN  
YLIOPISTO**  
UNIVERSITY  
OF TURKU

# **UNDERSTANDING THE STRUCTURE AND MEANING OF FINNISH TEXTS: FROM CORPUS CREATION TO DEEP LANGUAGE MODELLING**

---

Jenna Kanerva

## University of Turku

---

Faculty of Technology  
Department of Computing  
Computer Science  
Doctoral Programme in Technology

## Supervised by

---

Professor Tapio Salakoski  
Department of Mathematics  
and Statistics  
Faculty of Science  
University of Turku

Professor Veronika Laippala  
School of Languages and  
Translation Studies  
Faculty of Humanities  
University of Turku

## Reviewed by

---

Assistant Professor Miikka Silfverberg  
University of British Columbia, Canada

Assistant Professor Miryam de Lhoneux  
KU Leuven, Belgium

## Opponent

---

Associate professor Kairit Sirts  
University of Tartu, Estonia

The originality of this publication has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service.

Language technology tools utilizing artificial intelligence (AI) were used to proofread the language of this thesis. The usage of these tools follows the University of Turku guidelines.

ISBN 978-951-29-9622-3 (PRINT)  
ISBN 978-951-29-9623-0 (PDF)  
ISSN 2736-9390 (PRINT)  
ISSN 2736-9684 (ONLINE)  
Painosalama, Turku, Finland, 2024



UNIVERSITY OF TURKU

Faculty of Technology

Department of Computing

Computer Science

KANERVA, JENNA: Understanding the Structure and Meaning of Finnish Texts: From Corpus Creation to Deep Language Modelling

Doctoral dissertation, 190 pp.

Doctoral Programme in Technology

February 2024

## ABSTRACT

Natural Language Processing (NLP) is a cross-disciplinary field combining elements of computer science, artificial intelligence, and linguistics, with the objective of developing means for computational analysis, understanding or generation of human language. The primary aim of this thesis is to advance natural language processing in Finnish by providing more resources and investigating the most effective machine learning based practices for their use. The thesis focuses on NLP topics related to understanding the structure and meaning of written language, mainly concentrating on structural analysis (syntactic parsing) as well as exploring the semantic equivalence of statements that vary in their surface realization (paraphrase modelling). While the new resources presented in the thesis are developed for Finnish, most of the methodological contributions are language-agnostic, and the accompanying papers demonstrate the application and evaluation of these methods across multiple languages.

The first set of contributions of this thesis revolve around the development of a state-of-the-art Finnish dependency parsing pipeline. Firstly, the necessary Finnish training data was converted to the Universal Dependencies scheme, integrating Finnish into this important treebank collection and establishing the foundations for Finnish UD parsing. Secondly, a novel word lemmatization method based on deep neural networks is introduced and assessed across a diverse set of over 50 languages. And finally, the overall dependency parsing pipeline is evaluated on a large number of languages, securing top ranks in two competitive shared tasks focused on multilingual dependency parsing. The overall outcome of this line of research is a parsing pipeline reaching state-of-the-art accuracy in Finnish dependency parsing, the parsing numbers obtained with the latest pre-trained language models approaching (at least near) human-level performance.

The achievement of large language models in the area of dependency parsing — as well as in many other structured prediction tasks— brings up the hope of the large pre-trained language models genuinely comprehending language, rather than merely relying on simple surface cues. However, datasets designed to measure semantic comprehension in Finnish have been non-existent, or very scarce at the best. To address this limitation, and to reflect the general change of emphasis in the field towards task more semantic in nature, the second part of the thesis shifts its focus to language understanding through an exploration of paraphrase modelling. The second contribution of the thesis is the creation of a novel, large-scale, manually annotated

corpus of Finnish paraphrases. A unique aspect of this corpus is that its examples have been manually extracted from two related text documents, with the objective of obtaining non-trivial paraphrase pairs valuable for training and evaluating various language understanding models on paraphrasing. We show that manual paraphrase extraction can yield a corpus featuring pairs that are both notably longer and less lexically overlapping than those produced through automated candidate selection, the current prevailing practice in paraphrase corpus construction. Another distinctive feature in the corpus is that the paraphrases are identified and distributed within their document context, allowing for richer modelling and novel tasks to be defined.

**KEYWORDS:** NLP, Finnish, syntactic parsing, paraphrase modelling, machine learning

TURUN YLIOPISTO

Teknillinen tiedekunta

Tietotekniikan laitos

Tietojenkäsittelytieteet

KANERVA, JENNA: Understanding the Structure and Meaning of Finnish Texts: From Corpus Creation to Deep Language Modelling

Väitöskirja, 190 s.

Teknologian tohtoriorjelma

Helmikuu 2024

## TIIVISTELMÄ

Kieliteknologia on poikkiteollinen ala, joka yhdistää tietojenkäsittelytieteitä, tekoälyä ja kielitiedettä tavoitteenaan kehittää menetelmiä ihmisen käyttämän kielen laskennalliseen analysointiin, ymmärtämiseen tai tuottamiseen. Väitöstudiumukseni pyrin edistämään suomen kielelle tarjolla olevia kieliteknologian ratkaisuja tuottamalla uusia suomenkielisiä aineistoresursseja ja tutkimalla koneoppimiseen perustuvia menetelmiä niiden tehokkaaseen hyödyntämiseen. Väitöstudiumukseni keskittyy kirjoitetun kielen rakenteen ja merkityksen ymmärtämiseen, erityisesti rakenteelliseen analyysiin (syntaktinen jäsentäminen) ja sanamuodoiltaan erilaisten lausumien semanttisen vastaavuuden tutkimiseen (parafrasien mallintaminen). Vaikka väitöstudiumukseni puitteissa tuotetut aineistoresurssit on kehitetty nimenomaan suomen kieltä varten, useimmat esitellyistä menetelmistä ja työkaluista ovat kieliriippumattomia. Väitöstudiumukseeni sisältyvissä tutkimusartikkeleissa näitä menetelmiä onkin usein sovellettu ja arvioitu monilla eri kielillä.

Väitöstudiumukseni ensimmäinen osa-alue keskittyy koneoppimiseen perustuvan syntaktisen jäsentimen kehittämiseen ja sen arviointiin suomenkielisellä testiaineistolla. Kehittäminen alkaa koneoppimisen kannalta olennaisen koulutusaineiston, puupankin, muuntamisella Universal Dependencies (UD) -annotointijärjestelmän mukaiseksi. Tämän muunnoksen myötä suomenkielinen puupankki integroitiin osaksi kansainvälisesti tunnettua, monikielistä puupankkikokoelmaa, mikä loi perustan suomenkieliselle UD-jäsentämiselle. Seuraavaksi jäsentämiseen liittyvä tutkimukseni keskittyy uuden, syviin neuroverkkoihin perustuvan sanojen perusmuotoistamismenetelmän kehittämiseen ja arviointiin. Menetelmän osoitetaan olevan kilpailukykyinen yli 50 eri kielellä. Lopuksi kehitettyä jäsenintä, joka sisältää tekstin segmentoinnin, perusmuotoistamisen, morfologisen analyysin sekä sanojen riippuvuussuhteiden analysoinnin, arvioidaan useilla kielillä kahden monikieliseen jäsentämiseen keskittyvän shared task -kilpailun kontekstissa. Väitöstudiumukseni syntaktiseen jäsentämiseen keskittyvä lopputulema on koneoppittu, suurten kielimallien pohjalta toteutettu jäsenintä, jonka on tarkkuuden puolesta osoitettu yltävän (ainakin lähes) ihmistasoiseen suorituskykyyn suomenkielisellä testiaineistolla.

Suurten kielimallien tarkkuutta parantavat vaikutukset syntaktisessa jäsentämisessä — kuten myös monissa muissa rakenteellisissa ennustustehtävissä — herättävät toiveita siitä, että esikoulutetut kielimallit todella ymmärtävät tekstin merkityksen pelkkien pintapuolisten piirteiden sijaan. Tämän mittaaminen on kuitenkin



ollut haastavaa, sillä suomen kielen semanttisen ymmärtämisen arviointiin suunniteltuja korpuksia ei ole juurikaan ollut saatavilla. Tämän puutteen korjaamiseksi väitöstutkimukseni toinen osa-alue keskittyy rakenteen sijaan tekstin merkityksen ymmärtämiseen parafrasimallinnuksen kautta. Tämän osa-alueen päätavoitteena on luoda uudenlainen, käsin annotoitu suomenkielinen korpus parafrasien mallintamista varten. Korpuksen ainutlaatuinen piirre on, että se sisältää esimerkkejä, jotka on poimittu käsin kahdesta samankaltaisesta tekstidokumentista, esimerkiksi eri uutisartikkeleista, jotka kuvaavat samaa tapahtumaa. Yleinen parafrasikorpuksien koostamismenetelmä on ollut parafrasiparien koneellinen tunnistaminen ja käsintarkastus. Meidän manuaalinen menetelmämme pyrki löytämään haastavia esimerkkejä, joita on koneellisesti vaikeita tunnistaa, ja jotka ovat arvokkaita parafrasien ymmärtämismallien kehittämisen ja arvioinnin kannalta. Osoitamme, että käsin tehdyllä parafrasien poiminnalla voidaan saada aikaan korpus, joka sisältää pidempiä ja pintamuodoltaan vähemmän samankaltaisia pareja kuin aiemmat korpuksset. Korpuksemme toinen erityispiirre on, että parafrasiparit sisältävät luonnollisen kontekstinsa, mikä mahdollistaa monipuolisemman mallintamisen ja uusien koneoppimistehäviöiden määrittelyn.

ASIASANAT: kieliteknologia, suomi, syntaktinen jäsentäminen, parafrasien mallintaminen, koneoppiminen

# Acknowledgements

There are many people to whom I would like to express my sincere gratitude; without them, I would not be here. First, I want to thank my supervisors, Tapio and Veronika, for patiently guiding me through this journey and waiting for me to finish writing the thesis. Tapio, I want to express my sincere thanks to you for providing a supportive and stable environment for my research, overseeing the process, and being present whenever I needed something. Veronika, I am profoundly grateful to you for your interdisciplinary view and the broader perspective of the field you have introduced to me. I have learned a lot from you. Moreover, this journey has allowed me to get to know you not just as a supervisor but also as a friend, for which I am deeply grateful.

Filip, although you were not officially my supervisor, you are the person who has had the greatest impact on who I am today, at least professionally. You were the one who had the courage to hire me for the first time when I was just an inexperienced bachelor's student, and you were the one who taught me how to code. You have always had faith in me. Thank you for investing your time and patience in my work, I will forever be grateful for your guidance and friendship.

Next, I would like to extend my thanks to the reviewers of this thesis, Miikka Silfverberg and Miryam de Lhoneux, for their valuable feedback and suggestions. Additionally, I would like to express my gratitude to Kairit Sirts, who kindly agreed to act as my opponent.

I also want to thank all past and present members of the TurkuNLP research group; you have made the daily research environment lively and a pleasant place to work. Over the years, I have collaborated with dozens of people to all of whom I'm ultimately thankful. However, there are too many to address all by name. From my early years, I am especially grateful to Katri, who coordinated the very first annotation project I was involved in, as well as my fellow PhD students at that time, Juhani, Farrokh, Kai and Suwisa. Additionally, I would like to express my gratitude to Sampo, who has co-authored several publications with me and whose ambitions of pursuing scientific goals I value, and to Li-Hsin for working with me as a fellow PhD student during the latter years of my PhD journey.

I would also like to acknowledge the funding that has helped me achieve my goals over these years: the University of Turku Graduate School (UTUGS), the Turku University Foundation, and the Nokia Foundation. Additionally, I am grateful for the funding provided by the senior members of TurkuNLP, especially Filip, through his

numerous research projects.

I would also like to thank my family for their love and support. Iskä, Henni, Tarja-mummu ja Lasse-pappa, te ootte luonu elämälle perustan, jonka päälle on ollu helppo rakentaa. Kiitos teidän tinkimättömästä tuesta ja rohkaisusta, sekä niistä elämän viisauksista, joita ootte omalla esimerkillä vuosien varrella jakanu. I also want to acknowledge those who are not here anymore to see me finishing this journey; I owe you so much. Finally, I would like to thank my children, Aleksis and Amanda, along with the extended family, for giving me the correct perspective on life and showing me what really matters.

February 2024  
*Jenna Kanerva*

# Table of Contents

<b>Acknowledgements</b> . . . . .	<b>viii</b>
<b>Table of Contents</b> . . . . .	<b>x</b>
<b>Abbreviations</b> . . . . .	<b>xii</b>
<b>List of Original Publications</b> . . . . .	<b>xiv</b>
<b>Main contributions</b> . . . . .	<b>xv</b>
<b>Own contributions</b> . . . . .	<b>xvi</b>
<b>List of Co-Authored Publications not Included in the Thesis</b>	<b>xvii</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Natural Language and Human Language Technology . . . . .	1
1.2 Machine Learning . . . . .	2
1.3 NLP Tasks Relevant to the Thesis . . . . .	4
1.3.1 Tokenization . . . . .	5
1.3.2 Sentence Splitting . . . . .	6
1.3.3 Lemmatization . . . . .	6
1.3.4 Part-of-Speech and Morphological Tagging . . . . .	7
1.3.5 Dependency Parsing . . . . .	8
1.3.6 Paraphrase modelling . . . . .	10
1.4 Corpus Annotation . . . . .	11
1.4.1 Turku Dependency Treebank . . . . .	13
1.4.2 Universal Dependencies . . . . .	14
1.4.3 Finnish Paraphrase Corpus . . . . .	15
1.5 Research Objectives and Disposition of the Thesis . . . . .	16
<b>2 From Raw Text to Dependency Graphs</b> . . . . .	<b>19</b>
2.1 Turku Dependency Treebank into Universal Dependencies . . . . .	20
2.1.1 Overall Approach . . . . .	21
2.1.2 Part-of-Speech and Morphological Features . . . . .	22

2.1.3	Lemmas . . . . .	23
2.1.4	Dependency Relations . . . . .	24
2.1.5	Discussion and Outcome . . . . .	26
2.2	Background on Statistical Parsing of Finnish . . . . .	27
2.2.1	Pre-Neural Times . . . . .	27
2.2.2	Early Steps in Neural Parsing of Finnish . . . . .	29
2.3	Sequence-to-Sequence Lemmatizer for Finnish . . . . .	30
2.3.1	On the Contextuality of Lemmatization . . . . .	31
2.3.2	Modelling Lemmatization Using Sequence-to-Sequence Framework . . . . .	32
2.3.3	Results and Discussion . . . . .	34
2.4	Turku Neural Parser Pipeline . . . . .	35
2.4.1	Parser Modules . . . . .	36
2.4.2	Turku Neural Parser Pipeline at CoNLL-18 Shared Task . . . . .	38
2.5	Turku Enhanced Parser Pipeline . . . . .	39
2.5.1	Pre-Trained Contextualized Language Models . . . . .	39
2.5.2	Enhanced Graph Representation . . . . .	41
2.5.3	Turku Enhanced Parser Pipeline at the IWPT-2020 Shared Task . . . . .	44
2.6	Parser Utilization and Discussion . . . . .	46
<b>3</b>	<b>From Structure to Meaning . . . . .</b>	<b>49</b>
3.1	Building the Turku Paraphrase Corpus . . . . .	50
3.1.1	Manual Paraphrase Extraction . . . . .	50
3.1.2	Paraphrase Annotation . . . . .	54
3.2	Corpus Statistics and Evaluation . . . . .	57
3.3	Discussion and Paraphrase Modelling Experiments . . . . .	61
3.3.1	Paraphrase Classification . . . . .	61
3.3.2	Sentence Embeddings . . . . .	62
3.3.3	Paraphrase Span Retrieval . . . . .	64
<b>4</b>	<b>Conclusions and Future Work . . . . .</b>	<b>66</b>
	<b>List of References . . . . .</b>	<b>71</b>
	<b>Original Publications . . . . .</b>	<b>85</b>

# Abbreviations

BERT	Bidirectional Encoder Representations from Transformers (model)
BLEX	Bi-Lexical dependency score
CLAS	Content-word Labeled Attachment Score
CoNLL	Conference on Computational Natural Language Learning
CRF	Conditional Random Fields
DAG	Directed Acyclic Graph
ELAS	Labeled Attachment Score on Enhanced dependencies
ELECTRA	Efficiently Learning an Encoder that Classifies Token Replacements Accurate (model)
ELMo	Embeddings from Language Model (model)
FinBERT	Finnish BERT (model)
FST	Finite State Transducer
GPT	Generative Pre-trained Transformer (model)
GRU	Gated Recurrent Unit
IAA	Inter-Annotator Agreement
IWPT	International Workshop on Parsing Technologies
LAS	Labeled Attachment Score
LSTM	Long-Short Term Memory
mBERT	Multilingual BERT (model)
MLAS	Morphology-aware Labeled Attachment Score
MRPC	Microsoft Research Paraphrase Corpus
MST	Maximum Spanning Tree
MTW	MultiToken Word
MWE	MultiWord Expression
MWT	MultiWord Token
NLP	Natural Language Processing
OMorFI	Open Morphology of Finnish (model)
POS	Part-Of-Speech
QA	Question Answering
QQP	Quora Question Pairs
ReLU	Rectified Linear Unit
SBERT	Sentence-BERT (model)
SD	Stanford Dependencies

T5	Text-To-Text Transfer Transformer (model)
TDT	Turku Dependency Treebank
UAS	Unlabeled Attachment Score
UD	Universal Dependencies
ULMFit	Universal Language Model Fine-tuning (model)
XLM-R	Cross-lingual Language Model — Robustly Optimized BERT Pre-training Approach (model)

# List of Original Publications

This dissertation is based on the following original publications, which are referred to in the text by their Roman numerals:

- I Sampo Pyysalo, Jenna Kanerva, Anna Missilä, Veronika Laippala, and Filip Ginter. Universal Dependencies for Finnish. In Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA). 2015.
- II Jenna Kanerva, Filip Ginter, and Tapio Salakoski. Universal Lemmatizer: A Sequence to Sequence Model for Lemmatizing Universal Dependencies Treebanks. *Natural Language Engineering*. 2021; 27(5):545–574.
- III Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. Turku Neural Parser Pipeline: An End-to-End System for the CoNLL 2018 Shared Task. In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. 2018.
- IV Jenna Kanerva, Filip Ginter, and Sampo Pyysalo. Turku Enhanced Parser Pipeline: From Raw Text to Enhanced Graphs in the IWPT 2020 Shared Task. In Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies. 2020.
- V Jenna Kanerva, Filip Ginter, Li-Hsin Chang, Iiro Rastas, Valtteri Skantsi, Jemina Kilpeläinen, Hanna-Mari Kupari, Aurora Piirto, Jenna Saarni, Maija Sevón, and Otto Tarkka. Towards Diverse and Contextually Anchored Paraphrase Modeling: A Dataset and Baselines for Finnish. *Natural Language Engineering*. Published online 2023; 1-35.

The original publications have been reproduced with the permission of the copyright holders.



# Main contributions in the thesis

1. Conversion of the Turku Dependency Treebank (TDT) into Universal Dependencies (UD), its inclusion into the first release of the UD treebank collection, as well as continuous incorporation of the evolving UD guidelines into the treebank thereafter. (Paper I)
2. A novel approach to word lemmatization in context, utilizing the combination of character-level sequence-to-sequence models and morphological tags from the dependency parser. This method demonstrated superior performance, leading to the top rank in the lemmatization-sensitive BLEX score in the CoNLL shared task. (Paper II)
3. The development of a state-of-the-art dependency parser for Finnish along with other over fifty languages. This parser achieved top ranks in the CoNLL shared task, and together with pre-trained language models push the accuracy of Finnish dependency parsing into the 90+ LAS range. (Papers III and IV)
4. Support for enhanced dependencies in the parser, receiving the top rank in the International Workshop on Parsing Technologies (IWPT) shared task. (Paper IV)
5. The creation of a substantial corpus of Finnish paraphrases. Unlike in prior work, the paraphrases are manually extracted to avoid trivial pairs with high lexical overlap, as well as distributed and classified in the original document context. Initial paraphrase detection models were trained and evaluated on the data. (Paper V)

# Own contributions

- In **Paper I**, my primary responsibility was in managing the conversion of the syntactic trees, including manual refinement and validation. Further, following the initial data release through this paper, I have been maintaining all annotation layers of the treebank within the Universal Dependencies (UD) collection, i.e. incorporating to the data all the various changes to the Universal Dependencies annotation guidelines.
- **Paper II** was solely my work, from conceiving the idea and methodology, through implementing and analyzing the experiments, to drafting the manuscript.
- In **Paper III**, I led and coordinated the overall effort, coded a substantial portion of the pipeline, trained all models for all languages, and managed the submission of all parses to the shared task. I also wrote most of the manuscript.
- In **Paper IV**, I again led and coordinated the overall effort, created the system pipeline, trained all parser models, handled the shared task submission, and led the manuscript writing. This paper contains equal contribution by all co-authors.
- In **Paper V**, I led the entire corpus annotation process. This includes both the practical annotation process management, including coding many of the tools used in the annotation, as well as the definition of the task itself and the design of the annotation guidelines. I also played a key role in planning, executing and analyzing the experiments, and wrote the majority of the manuscript.

# List of Co-Authored Publications not Included in the Thesis

Below are listed, in reverse chronological order, other 47 co-authored peer-reviewed publications (in part under maiden name Nyblom) that are not included as a part of the doctoral thesis proper.

Jenna Kanerva, Hanna Kitti, Li-Hsin Chang, Teemu Vahtola, Mathias Creutz, Filip Ginter. Semantic Search as Extractive Paraphrase Span Detection. *Language Resources and Evaluation*, 2024.

Risto Luukkonen, Ville Komulainen, Jouni Luoma, Anni Eskelinen, Jenna Kanerva, Hanna-Mari Kristiina Kupari, Filip Ginter, Veronika Laippala, Niklas Muennighoff, Aleksandra Piktus, Thomas Wang, Nouamane Tazi, Teven Le Scao, Thomas Wolf, Osma Suominen, Samuli Sairanen, Mikko Merioksa, Jyrki Heinonen, Aija Vahtola, Samuel Antao, and Sampo Pyysalo. FinGPT: Large Generative Models for a Small Language. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Hannu Salmi, Jenna Kanerva, Harri Kiiskinen, and Filip Ginter. Paimen, piika ja emäntä: Arvot ja ammatit suomalaisessa näytelmäelokuvassa 1907–2017. *Lähikuva – audiovisuaalisen kulttuurin tieteellinen julkaisu*, 35(4):8–18, 12 2022.

Li-Hsin Chang, Jenna Kanerva, and Filip Ginter. Towards automatic short answer assessment for Finnish as a paraphrase retrieval. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, 2022.

Sebastian Gehrmann, Abhik Bhattacharjee, Abinaya Mahendiran, Alex Wang, Alexandros Papangelis, Aman Madaan, Angelina Mcmillan-major, Anna Shvets, Ashish Upadhyay, Bernd Bohnet, Bingsheng Yao, Bryan Wilie, Chandra Bhagavatula, Chaobin You, Craig Thomson, Cristina Garbacea, Dakuo Wang, Daniel Deutsch, Deyi Xiong, Di Jin, Dimitra Gkatzia, Dragomir Radev, Elizabeth Clark, Esin Durmus, Faisal Ladhak, Filip Ginter, Genta Indra Winata, Hendrik Strobelt, Hiroaki Hayashi, Jekaterina Novikova, Jenna Kanerva, Jenny Chim, Jiawei Zhou, Jordan Clive, Joshua Maynez, João Sedoc, Juraj Juraska, Kaustubh Dhole, Khyathi Raghavi Chandu, Laura Perez

Beltrachini, Leonardo F. R. Ribeiro, Lewis Tunstall, Li Zhang, Mahim Pushkarna, Mathias Creutz, Michael White, Mihir Sanjay Kale, Moussa Kamal Eddine, Nico Daheim, Nishant Subramani, Ondrej Dusek, Paul Pu Liang, Pawan Sasanka Ammanamanchi, Qi Zhu, Ratish Puduppully, Reno Kriz, Rifat Shahriyar, Ronald Cardenas, Saad Mahamood, Salomey Osei, Samuel Cahyawijaya, Sanja Štajner, Sebastien Montella, Shailza Jolly, Simon Mille, Tahmid Hasan, Tianhao Shen, Tosin Adewumi, Vikas Raunak, Vipul Raheja, Vitaly Nikolaev, Vivian Tsai, Yacine Jernite, Ying Xu, Yisi Sang, Yixin Liu, and Yufang Hou. GEMv2: Multilingual NLG Benchmarking in a Single Line of Code. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2022.

Filip Ginter, Harri Kiiskinen, Jenna Kanerva, Li-Hsin Chang, and Hannu Salmi. Deep Learning and film history: Model explanation techniques in the analysis of temporality in Finnish fiction film metadata. In *Proceedings of the 6th Digital Humanities in the Nordic and Baltic Countries Conference (DHNB 2022)*, 2022.

Jenna Kanerva and Filip Ginter. Out-of-domain evaluation of Finnish dependency parsing. In *Proceedings of the 13th International Conference on Language Resources and Evaluation (LREC'22)*, pages 1114–1124, 2022.

Li-Hsin Chang, Sampo Pyysalo, Jenna Kanerva, and Filip Ginter. Quantitative evaluation of alternative translations in a corpus of highly dissimilar Finnish paraphrases. In *Proceedings for the First Workshop on Modelling Translation: Translatology in the Digital Age*, 2021.

Sampo Pyysalo, Jenna Kanerva, Antti Virtanen, and Filip Ginter. WikiBERT models: Deep transfer learning for many languages. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021)*, 2021.

Jenna Kanerva, Filip Ginter, Li-Hsin Chang, Iiro Rastas, Valtteri Skantsi, Jemina Kilpeläinen, Hanna-Mari Kupari, Jenna Saarni, Maija Sevón, and Otto Tarkka. Finnish paraphrase corpus. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021)*, 2021.

Jörg Tiedemann, Tommi Nieminen, Mikko Aulamo, Jenna Kanerva, Akseli Leino, Filip Ginter, and Niko Papula. The FISKMÖ Project: Resources and tools for Finnish-Swedish machine translation and cross-linguistic research. In *Proceedings of 12th Conference on Language Resources and Evaluation LREC'2020*, 2020.

Jenna Kanerva, Filip Ginter, and Sampo Pyysalo. Dependency parsing of biomedical text with BERT. *BMC Bioinformatics*, 21(23):1–12, 2020.

Thang Minh Ngo, Jenna Kanerva, Filip Ginter, and Sampo Pyysalo. Neural dependency parsing of biomedical text: TurkuNLP entry in the CRAFT structural annotation task. In *Proceedings of the BioNLP Open Shared Tasks 2019*, 2019.

Jenna Kanerva, Samuel Rönqvist, Riina Kekki, Tapio Salakoski, and Filip Ginter. Template-free data-to-text generation of finnish sports news. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa'19)*, 2019.

Samuel Rönqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. Is multilingual BERT fluent in language generation? In *Proceedings of the 1st NLPL Workshop on Deep Learning for Natural Language Processing*, 2019.

Kira Drozanova, Daniel Zeman, Jenna Kanerva, and Filip Ginter. Parse Me if You Can: Artificial Treebanks for Parsing Experiments on Elliptical Constructions. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018a. European Language Resources Association (ELRA).

Veronika Laippala, Aki-Juhani Kyröläinen, Jenna Kanerva, and Filip Ginter. Dependency profiles in the large-scale analysis of discourse connectives. *Corpus linguistics and linguistic theory*, 2018.

Joakim Nivre, Paola Marongiu, Filip Ginter, Jenna Kanerva, Simonetta Montemagni, Sebastian Schuster, and Maria Simi. Enhancing Universal Dependency treebanks: A case study. In *Proceedings of the 2018 Workshop on Universal Dependencies (UDW 2018)*. Association for Computational Linguistics, 2018.

Kira Drozanova, Filip Ginter, Jenna Kanerva, and Daniel Zeman. Mind the gap: Data enrichment in dependency parsing of elliptical constructions. In *Proceedings of the 2018 Workshop on Universal Dependencies (UDW 2018)*. Association for Computational Linguistics, 2018b.

Juhani Luotolahti, Jenna Kanerva, and Filip Ginter. Dep\_search: Efficient search tool for large dependency parsebanks. In *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa)*, Gothenburg, Sweden, 2017a. Linköping University Electronic Press.

Jenna Kanerva, Juhani Luotolahti, and Filip Ginter. TurkuNLP: Delexicalized pre-training of word embeddings for dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, 2017a.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gökırmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Uřešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Drozanova, Héctor Martínez Alonso, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim

Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadova, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, 2017.

Jenna Kanerva, Sampo Pyysalo, and Filip Ginter. Fully delexicalized contexts for syntax-based word embeddings. In *Proceedings of the International Conference on Dependency Linguistics (Depling'17)*, 2017b.

Marie-Catherine de Marneffe, Matias Gironi, Jenna Kanerva, and Filip Ginter. Assessing the annotation consistency of the Universal Dependencies corpora. In *Proceedings of the International Conference on Dependency Linguistics (Depling'17)*, 2017.

Juhani Luotolahti, Jenna Kanerva, and Filip Ginter. Cross-lingual pronoun prediction with deep recurrent neural networks v2.0. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 63–66, Copenhagen, Denmark, September 2017b. Association for Computational Linguistics.

Tuomas Huomo, Aki-Juhani Kyröläinen, Jenna Kanerva, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Veronika Laippala. Distributional semantics of the partitive A argument construction in Finnish. *Empirical approaches to cognitive linguistics: Analysing real-life data*, 2017.

Jörg Tiedemann, Fabienne Cap, Jenna Kanerva, Filip Ginter, Sara Stymne, Robert Östling, and Marion Weller-Di Marco. Phrase-based smt for Finnish with more data, better models and alternative alignment and translation tools. In *Proceedings of the First Conference on Machine Translation*, pages 391–398, Berlin, Germany, August 2016. Association for Computational Linguistics.

Veronika Laippala, Aki-Juhani Kyröläinen, Jenna Kanerva, Juhani Luotolahti, and Filip Ginter. Dependency profiles as a tool for big data analysis of linguistic constructions: A case study of emoticons. *Journal of Estonian and Finno-Ugric Linguistics. Grammar in Use: Approaches to Baltic Finnic.*, 8:127–153, 2017.

Juhani Luotolahti, Jenna Kanerva, and Filip Ginter. Cross-lingual pronoun prediction with deep recurrent neural networks. In *Proceedings of the First Conference on Machine Translation*, pages 596–601, Berlin, Germany, August 2016. Association for Computational Linguistics.

Jenna Kanerva, Juhani Luotolahti, and Filip Ginter. Turku: Semantic dependency parsing as a sequence classification. In *Proceedings of SemEval 2015*, pages 265–269. Association for Computational Linguistics, 2015.

Veronika Laippala, Jenna Kanerva, Sampo Pyysalo, Anna Missilä, Tapio Salakoski, and Filip Ginter. Syntactic N-grams in the classification of the Finnish Internet Parsebank: Detecting translations and informality. In *Proceedings of NoDaLiDa 2015*, pages 108–116. NEALT, 2015a.

Juhani Luotolahti, Jenna Kanerva, Veronika Laippala, Sampo Pyysalo, and Filip Ginter. Towards universal web parsebanks. In *Proceedings of the International Conference on Dependency Linguistics (Depling'15)*, pages 211–220. Uppsala University, 2015a.

Juhani Luotolahti, Jenna Kanerva, Sampo Pyysalo, and Filip Ginter. SETS: scalable and efficient tree search in dependency graphs. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 51–55. Association for Computational Linguistics, 2015b.

Katri Haverinen, Jenna Kanerva, Samuel Kohonen, Anna Missilä, Stina Ojala, Timo Viljanen, Veronika Laippala, and Filip Ginter. The Finnish Proposition Bank. *Language Resources and Evaluation*, 49(4):907–926, 2015.

Veronika Laippala, Jenna Kanerva, and Filip Ginter. Syntactic Ngrams as keystuctures reflecting typical syntactic patterns of corpora in Finnish. *Procedia – Social and Behavioral Sciences. Current Work in Corpus Linguistics: Working Traditionally-conceived Corpora and Beyond. Selected Papers from the 7th International Conference on Corpus Linguistics (CILC2015)*, 198:233–241, 2015b.

Jörg Tiedemann, Filip Ginter, and Jenna Kanerva. Morphological segmentation and OPUS for Finnish-English machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 177–183, Lisbon, Portugal, 2015. Association for Computational Linguistics.

Jenna Kanerva and Filip Ginter. Post-hoc manipulations of vector space models with application to Semantic Role Labeling. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) at EACL'14*, pages 1–10, 2014.

Jenna Kanerva, Juhani Luotolahti, and Filip Ginter. Turku: Broad-coverage semantic parsing with rich features. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 678–682. Association for Computational Linguistics and Dublin City University, 2014a.

Jenna Kanerva, Matti Luotolahti, Veronika Laippala, and Filip Ginter. Syntactic N-gram collection from a large-scale corpus of internet Finnish. In *Proceedings of the Sixth International Conference Baltic HLT 2014*, pages 184–191. IOS Press, 2014b.

Veronika Laippala, Timo Viljanen, Antti Airola, Jenna Kanerva, Sanna Salanterä, Tapio Salakoski, and Filip Ginter. Statistical parsing of varieties of clinical Finnish.

*Artificial Intelligence in Medicine*, 61(3):131–136, 2014. ISSN 0933-3657. Text Mining and Information Analysis of Health Documents.

Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. Building the essential resources for Finnish: the Turku Dependency Treebank. *Language Resources and Evaluation*, 48:493–531, 2014. ISSN 1574-020X. Open access.

Katri Haverinen, Filip Ginter, Veronika Laippala, Samuel Kohonen, Timo Viljanen, Jenna Nyblom, and Tapio Salakoski. A dependency-based analysis of treebank annotation errors. In Kim Gerdes, Eva Hajičová, and Leo Wanner, editors, *Computational Dependency Theory*. IOS Press, 2013a.

Filip Ginter, Jenna Nyblom, Veronika Laippala, Samuel Kohonen, Katri Haverinen, Simo Vihjanen, and Tapio Salakoski. Building a large automatically parsed corpus of Finnish. In *Proceedings of the 19th Nordic Conference on Computational Linguistics (NoDaLiDa'13)*, pages 291–300, 2013.

Katri Haverinen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Jenna Nyblom, Stina Ojala, Timo Viljanen, Tapio Salakoski, and Filip Ginter. Towards a dependency-based PropBank of general Finnish. In *Proceedings of the 19th Nordic Conference on Computational Linguistics (NoDaLiDa'13)*, pages 41–57, 2013b.

Veronika Laippala, Timo Viljanen, Antti Airola, Jenna Nyblom, Sanna Salanterä, Tapio Salakoski, and Filip Ginter. Statistical parsing of varieties of clinical Finnish. In Hanna Suominen, editor, *Proceedings of the 4th International Louhi Workshop on Health Document Text Mining and Information Analysis (Louhi 2013)*, 2013.

Jenna Nyblom, Samuel Kohonen, Katri Haverinen, Tapio Salakoski, and Filip Ginter. Predicting conjunct propagation and other extended Stanford Dependencies. In *Proceedings of the International Conference on Dependency Linguistics (Depling 2013)*, pages 252–261, 2013.

Katri Haverinen, Filip Ginter, Veronika Laippala, Samuel Kohonen, Timo Viljanen, Jenna Nyblom, and Tapio Salakoski. A dependency-based analysis of treebank annotation errors. In *Proceedings of International Conference on Dependency Linguistics (Depling'11)*, Barcelona, Spain, pages 115–124, 2011.



# 1 Introduction

## 1.1 Natural Language and Human Language Technology

A natural way for humans to communicate is through language. While much of the daily interactions occur verbally, written language is particularly effective for information storage and non-real-time communication, the real-time text communication also increasing its popularity through modern instant messaging platforms. Consequently, in modern societies a significant proportion of information produced daily is transmitted and stored as digital text, common everyday use including e.g. digital newspapers and books, user manuals and administrative forms, as well as text messages, and web pages. However, the mere existence of textual data is insufficient for knowledge preservation and transmission; transmitting information and knowledge requires the recipient to understand and “decode” the meaning of the text — a process known as reading and comprehending. While each language includes a certain set of common practices (grammar) to ensure a smooth process of communication, decoding the intended meaning from human language is not always a straightforward task, especially if automatic computer processing is involved. With the current volumes of existing textual data, automatic computer processing is crucial for making information exchange and storage more efficient and accessible.

In **Natural Language Processing** (NLP), computers are employed to perform tasks involving human language (Jurafsky and Martin, 2009), ranging from simple word counting in text editors to more complex tasks requiring a deep understanding of language, for instance machine translation or question answering. This thesis tackles NLP topics related to understanding the structure and meaning of language, mainly concentrating on structural analysis of language (**syntactic parsing**) as well as understanding the deeper meaning of text beyond its structure, especially studying the equivalence in meaning of statements differing in their surface realization (**paraphrase modelling**). On the methodological aspect, the thesis addresses topics from corpus creation to deep learning models. While the methodological contributions introduced in the thesis are language-agnostic and therefore not optimized for any specific language, the motivation of the work is heavily inspired by the Finnish language. The overall objective of the thesis is to advance natural language processing in Finnish by providing more resources and exploring best machine learning based practices in their utilization. Therefore, the thesis has a general goal of sup-

porting the local language usage by improving the situation of technologies available for the language users. In addition, Finnish is a morphologically complex language from a non-Indo-European language family, serving as an interesting example of a less-resourced language in the field where a lot of research still focuses on English. We hope that our work on Finnish can benefit also other languages that share similar characteristics.

The publications included into the thesis are dated between 2015 to 2023, covering the main contributions of developing a neural parsing pipeline for Finnish language using Universal Dependencies annotation framework, as well as developing a corpus of Finnish paraphrases and neural models for paraphrase classification. However, I have contributed to many topically related publications listed separately, including but not limited to construction of the Turku Dependency Treebank (Haverinen et al., 2014), construction of the Finnish Proposition Bank (Haverinen et al., 2015), construction of the Finnish Internet Parsebank (Kanerva et al., 2014; Luotolahti et al., 2015), Finnish text generation (Kanerva et al., 2019) and deep language modelling of Finnish (Virtanen et al., 2019; Rönnqvist et al., 2019; Pyysalo et al., 2021).

The structure of the thesis is as follows: First, a general background of the thesis topics is provided, introducing machine learning approaches, NLP tasks relevant to this thesis, as well as corpus annotation and annotated datasets. At the end of Chapter 1, the research objectives are outlined and their relation to the papers is discussed. Chapters 2 and 3 each address one broader topic of the thesis, the former concentrating on syntactic parsing while latter focuses on paraphrase modelling. Finally, Chapter 4 concludes the work, summarizing the key findings, linking them back to research questions, and suggesting ideas for future work. Instead of including dedicated sections for discussion and related work, the thesis integrates these aspects throughout the text.

## 1.2 Machine Learning

The methods presented in this thesis are based on machine learning. Machine learning methods are generally divided into two broad categories, supervised machine learning and unsupervised machine learning. In **supervised machine learning** the model is trained to predict a known output value based on the given input, while in **unsupervised learning** the desired output values are not known beforehand and the model is learning to induce general patterns from the data. While supervised learning is often seen as a more direct method to obtain targeted goals, the requirements for the availability of labeled training data poses its own challenges.

Modern neural network models, for example models based on the self-attention architecture (Vaswani et al., 2017), have shown huge improvements in many NLP related tasks (Wang et al., 2018, 2019). Such models are typically very large in

size, i.e. include a large number of parameters, thus also requiring a large amount of training data. To overcome the need of large amount of costly annotated, labeled data, many different transfer learning techniques are introduced (Ruder et al., 2019). In **transfer learning**, the model is first trained to solve a task, and the knowledge obtained during that training is later transferred to another task. In this setting, the hope is to utilize knowledge from a related task in order to solve an objective where you may not have an excessive amount of training data. In NLP, a common way to apply transfer learning is through **pre-training**, where the model parameters are first trained with a large amount of unlabeled data to obtain a general representation of a language, and later on the pre-trained model is fine-tuned into the specific task with smaller amount of labeled training data by straightforwardly continuing the training from the pre-trained parameters with new training data and task specific training objective.

Pre-training can be based on any training objective beneficial for language understanding or the task in question, the most popular methods lately resorting to inducing meaning of words or sentences from unlabeled data using language modelling objective (Ruder et al., 2019). In **language modelling**, a supervised training objective can be used with unlabeled textual data, by simply predicting a masked word based on the observed context. The method thus only requires a large amount of raw text to learn how the language is formed from individual words, but no additional annotation is needed. A common approach to transfer learning using the language modelling objective is pre-trained word embeddings. **Word embeddings** are dense, continuous representations of words induced in such a manner where words with similar meanings or syntactic roles receive a representation similar to each other in mathematical terms. A common approach for inducing word embeddings is based on the distributional hypothesis, where the meaning of a word is characterized by the context it appears in (Harris, 1954; Firth, 1957), thus similar words appearing in similar contexts. The most common methods for inducing word embeddings based on the distributional hypothesis include word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and fasttext (Bojanowski et al., 2017) among others. While word embeddings are context independent, where each unique word receives exactly one static embedding irrespective of the context and the different meanings the word may have, contextualized language models induce a context dependent representation of text. In **contextual representation of text** the representation of a word depends on the actual context where the word appears, thus producing a different representation for the word *bank* in the contexts *bank account* and *river bank*. Popular methods for contextualized representations include neural language models such as ELMo (Peters et al., 2018), ULMFit (Howard and Ruder, 2018), BERT (Devlin et al., 2019), GPT (Radford et al., 2019), and ELECTRA (Clark et al., 2020).

Lately, the de-facto standard for training a supervised model for an NLP task has involved using pre-trained, contextualized language models, which are initially

pre-trained on a large amount of raw text and later fine-tuned to the specific task. In many benchmarks, such combination is shown to produce human parity or even super-human performance (Wang et al., 2019). Very recently, instruction fine-tuned generative models have had a substantial impact on the NLP field. Here, generative language models are aligned with user intent, i.e. to follow instruction given in the user’s input prompt (Ouyang et al., 2022). Such prompt-based models, e.g. GPT-4 (OpenAI, 2023), have shown impressive generalization capabilities on tasks not directly present in their training data. While such models are largely out of scope of this thesis, in Chapter 4 we briefly discuss the future of the thesis topics in relation to these models.

### 1.3 NLP Tasks Relevant to the Thesis

This section provides an overview of the NLP tasks relevant to the thesis. Many NLP applications are built as pipelines, where some degree of pre-processing is required before addressing the actual downstream objective, i.e. the task visible to the end users. Therefore, to address the overall objective, the system can be performing several relevant NLP tasks behind the scenes. The level of necessary pre-processing varies depending on the specific task and the chosen system design. With the increased popularity of the latest deep learning methods, the trend has shifted from heavily feature engineered pipelines towards end-to-end models directly producing the desired output based on the given raw text. Nevertheless, even with the recent end-to-end models, some intermediate processing standards may apply due to practical issues (such as memory or processing time constraints) or requirements posed by the task definition itself. For instance, machine translation as a task doesn’t inherently have any built-in requirements, however, many translation systems are still designed to process one sentence at a time and therefore, in practice requiring the text to be split into individual sentences before translation can occur. Pipeline design is beneficial especially with models where the complexity grows non-linearly with respect to the length of the text.

Firstly, we will introduce the common tasks involved in a structural analysis of the language, including text segmentation (tokenization and sentence splitting), lemmatization, morphological analysis (part-of-speech and morphological tagging), as well as syntactic analysis (dependency parsing). These steps are often incorporated into a single processing pipeline starting from raw text and continuing through intermediate analysis layers all the way to dependency relations, where the different tasks may be executed consecutively (predicting one task at a time in a specific order) or jointly (predicting two or more task simultaneously). Secondly, we introduce the paraphrase modelling as an NLP objective used to study the meaning behind the surface language.

### 1.3.1 Tokenization

Many NLP tasks are designed to operate on token-level, where token is a piece of text treated as single unit, typically a word or a punctuation character. The process of identifying token boundaries is called **tokenization** or **word segmentation**. While in many languages, individual words are separated using spacing, and the most simple tokenization method could rely on whitespace splitting, in certain cases the spacing is omitted (e.g. before punctuation characters in English) or not used at all (e.g. Chinese or Thai languages).

In addition to the basic tokenization, if comprehended as separating atomic units written together only due to conventional spelling rules, several extension are introduced in different studies. A **multiword token** (MWT) is a single orthographic token that corresponds to multiple syntactic words and thus, in syntactic analysis would receive two meaningful relations (Nivre et al., 2020). For example in Finnish, many coordinate and subordinate conjunctions can be merged with a negation verb creating an orthographically indivisible word, e.g. *muttei* 'but not', which at the same time serves as a conjunction and a negative auxiliary verb. (Hakulinen et al. (2014), §139–§141)

On the other hand, sometimes multiple orthographic tokens can represent single syntactic units without any internal structure (Kahane et al., 2017), which could sometimes be considered to be single syntactic words. Most common cases are **multiword expressions** (MWEs), which are well established phrases with a single syntactic function (such as *in spite of* in English or *mikä tahansa* 'what ever' in Finnish). Similar constructions include also **multitoken words** (MTWs), which are atomic units, such as numerical expressions, emoticons or abbreviations, occasionally written with spacing (such as *e. g., :*), or *100 000*), however, clearly constituting a single meaningful unit which could be written without spacing as well.

The tokenization work described in this thesis follows the syntax-oriented tokenization guidelines defined in the Universal Dependencies project (Nivre et al., 2016, 2020), a dependency annotation framework introduced in Section 1.4.2. In short, the UD tokenization guidelines support analysing multiword tokens (MWT) as separate syntactic tokens, as well as allowing a restricted set of phenomena to include whitespace inside a token (MTWs, e.g. emoticons and numeric expressions), however, multiword expressions (MWEs) are analyzed as separate syntactic words rather than word-like units.

Several methods for tokenization are suggested in the literature; the traditional methods often utilize regular expressions (see e.g. Manning et al. (2014)), while the latest state-of-the-art methods usually rely on supervised machine learning (see e.g. Zeman et al. (2018); Qi et al. (2020); Nguyen et al. (2021)). When casting the tokenization as a machine learning problem, the typical approach is to predict for each character whether it is an end-of-token character or not. Such approach can

further be restricted to certain characters only, for example preventing tokenizing inside a sequence of letters without interfering punctuation or spacing, if needed.

### 1.3.2 Sentence Splitting

**Sentence splitting** or **sentence segmentation** is the process of dividing running text into individual sentences to be able to process one sentence at a time. In many cases simple heuristics for recognizing sentence boundaries will apply, the most typical case being sentence-final punctuation (e.g. a dot, an explanation mark or a question mark) followed by a whitespace and an uppercased letter. However, there are several ambiguities with similar structures that do not imply sentence boundaries (e.g. *Mr. Smith*), as well as cases where these visible sentence boundary features are omitted.

Further, the materials processed in NLP are often obtained through automatic collection, sometimes producing artefacts associated with the collection methods used, such as web crawling or different data conversion techniques. Due to these methods, the text layout may not strictly follow the original, and different text segments, e.g. titles or paragraphs, may be merged together without retaining the visual spacing of the original layout. In practice, this means that separate, sentence-like items may be joined in the text. A typical example is the title of a news article being followed by the main body without any sentence ending marker, as HTML formatting is lost at some point in the processing pipeline. In such cases, sentence splitting methods not limited to orthographic writing rules are necessary.

Similarly to tokenization, the traditional methods for sentence splitting often utilize regular expression (see e.g. Manning et al. (2014)), however, latest methods usually rely on supervised machine learning (see e.g. Zeman et al. (2018); Qi et al. (2020); Nguyen et al. (2021)). When casting sentence segmentation as a machine learning problem, one of the typical approaches is to predict for each character or token whether it is a sentence-ending-marker or not. Sentence segmentation can easily be predicted jointly with the tokenization by simply using three classes instead of two, token-ending-marker, sentence-ending-marker, or no-boundary-marker, where sentence boundary naturally indicates also the end of the token.

### 1.3.3 Lemmatization

**Lemmatization** is a process of determining the **lemma** (also referred to as the base form or the dictionary form) for a given surface word appearing in a text. In many languages, the lemma is considered to be the singular nominative for nouns, the infinitive for verbs etc., thus in the process of lemmatization both *computer* and *computers* are transformed to their common lemma *computer*, and the words *is*, *am*, *are* and *were* to their common lemma *(to) be*. For languages possessing rich morphology — like Finnish — lemmatization is an important step for numerous downstream

applications.

Many words are ambiguous in terms of lemmatization, where for the same surface form multiple plausible lemmas exist, and the correct lemma must be understood from the context where the word appears. For example, the English word *lives* can be inflected from two distinct lemmas depending on the context, "(to) live" as in "*She lives in Finland.*" or "life" as in "*Cats are said to have multiple lives.*". Thus, when lemmatizing words in a running text, it is crucial to also understand the sentence structure and/or meaning in order to lemmatize ambiguous word forms correctly. Lemmatization methods can therefore be divided into two categories, context-aware methods where the lemmatization system is aware of the context, and methods where the system is lemmatizing individual words without contextual information. The advantage in the former approach is the ability to correctly lemmatize ambiguous words based on the contextual information while the latter is only able to either give one lemma for each word regarding its contextual meaning, or list all alternatives.

As lemmatization is defined as determining lemmas for individual words, tokenization is a required preprocessing step to first determine the token boundaries before lemmatizing the individual words. Lemmatizers can be implemented using rule-based methods relying on a lexicon and grammar rules of the language (see e.g. Pirinen (2008)), however, the current state-of-the-art methods often rely on supervised machine learning by either using edit-tree classifiers predicting which of the known edit-trees will produce a correct lemma for the given word (see e.g. Straka et al. (2016)), or sequence-to-sequence transformations generating the sequence of lemma characters for the given sequence of word form characters (see e.g. Paper II or Bergmanis and Goldwater (2018)).

### 1.3.4 Part-of-Speech and Morphological Tagging

Part-of-speech (POS) and morphological features describe the lexical categories the words belong to, as well as the lexical and grammatical properties they have. Generally, words belonging to the same part-of-speech group show similar syntactic behavior by often having the same grammatical role in the sentence as well as following similar inflectional patterns. In **morphological tagging**, for each word in the given text the practise is to assign one part-of-speech tag describing its grammatical category, and a set of morphological features describing its inflectional properties and other categorization features. For example, the English word *computers* is a noun inflected into its plural form, therefore receiving the *NOUN* part-of-speech tag and *Number=Plur* as morphological features, however the exact definitions and namings depends on the applied annotation guidelines.

Similar to lemmatization, and in general all task involving human language, also morphological tagging involves ambiguities. While some words have only one plausible set of morphological features, some words may have alternative analysis de-

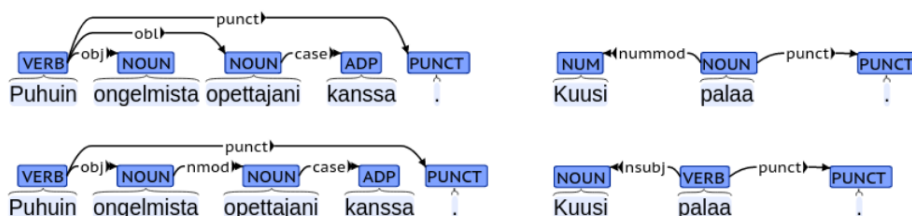
pending on the context, making these words morphologically ambiguous. Again, the English word form *lives* is one such word, including the possible analyses of *noun in plural* or *verb in 3rd person singular*. In this example, the word includes a mixture of morphological and lemmatization ambiguity as selecting one of the morphological analyses entails committing to the corresponding lemma as well, or the other way around. However, the ambiguity can be purely on morphological features as well, as is in the case of *run*, which can be e.g. a noun, an imperative verb, or a finite verb, each still receiving the same lemma analysis (if not taking into account the possible *to* in verb lemmas).

In order to introduce tags for each word, morphological tagging also requires tokenization as a preprocessing step in order to know the word boundaries. The classical approaches to morphological analysis of highly inflective languages are based on two-level morphology implemented using finite state transducers (FSTs) (Koskeniemi, 1984; Karttunen and Beesley, 1992). FSTs are models encoding vocabulary and string rewrite rules for analyzing an inflected word into its lemma and morphological tags. However, due to surface form ambiguity, the FST encodes all possible analyses for a word, and context-aware disambiguation methods are then needed in order to carry out context dependent morphological tagging. The disadvantage of morphological transducers is them relying on a list of lexical entries and failing to analyze a word not defined in the transducer's lexicon, however, this challenge can be (at least) partially addressed by the introduction of morphological guessers (Lindén, 2008), which are probabilistic models able to guess the potential analyses for such words. The state-of-the-art again relies on supervised machine learning, where sequence tagging is often applied. While the prediction of part-of-speech tags is quite straightforward, each word in general receiving exactly one tag, there are several techniques for predicting morphological features, where a set of appropriate tags needs to be predicted, predicting either all features at once as one complete tag, or each feature as a separate class (see e.g. Paper III).

### 1.3.5 Dependency Parsing

**Parsing**, also called automatic syntactic analysis, is the task of determining the syntactic structure of the sentence. In **dependency parsing**, the syntax is described in terms of dependencies, directed and labeled relations between words encoding the syntactic structure of the sentence, as well as the syntactic role each word holds. Each directed relation thus has a head token (also called governor) and a dependent token, and the relation can be read as the head token governing the dependent, or the dependent token depending on the head. Many of the dependency syntax frameworks require the sentence relations to form a tree structure (dependency tree), where each token can be a dependent of another by exactly once, and exactly one token is the sentence root not dependent of any token, while directly or indirectly govern-





**Figure 1.** An illustration of two ambiguous sentences with two plausible dependency analysis for both. In the first sentence, the attachment of the prepositional phrase is ambiguous, while the second sentence involves two ambiguous words creating two different interpretations of the sentence. The English translation for the first sentence reads *I talked about the problems with my teacher*, while the second can be translated *Spruce is on fire*, or *Six pieces*, depending on the intended meaning.

ing all other tokens in the sentence. The task in dependency parsing is then to find the 'correct' dependency tree for the sentence among all structurally possible, but meaningless or contextually inappropriate trees.

In case of structural or attachment ambiguity, the sentence has several 'correct' dependency trees attaching a word or words differently, and each correct dependency tree thus imply different meaning of the sentence. For example, sometimes the attachment decision of prepositional phrases adjusts the sentence meaning. In Figure 1 we illustrate two ambiguous sentences with two different dependency trees for both, the first one with prepositional phrase attachment ambiguity, where different interpretations changes the dependency tree structure, and the second with words involving lexical ambiguities, making it either a verb phrase or a noun phrase.

While it's typical to require syntactic relations producing a tree structure for the sentence, also extended layers of analysis are defined in many dependency formalisms producing dependency graphs rather than trees (Nivre et al., 2016; De Marneffe and Manning, 2008a). In these extended layers, the idea is to support explicitly marking secondary relations not supported by the base tree analysis, such as external subjects or conjunct propagation, where the same word receive two or more incoming relations.

As dependencies are defined as relations between syntactic words in a sentence, dependency parsing often relies on both sentence splitting as well as tokenization as necessary pre-processing steps. While many technical implementations are able to utilize morphological features and/or lemmas as additional features while parsing, these intermediate step are only optional and not strictly required. The state-of-the-art methods for dependency parsing rely on supervised machine learning. The machine learned methods for dependency parsing are often divided into two approaches: transition-based and graph-based parsing (Nivre, 2004; McDonald et al., 2005). In transition-based parsing, the construction of the dependency tree is defined as a sequence of strictly defined actions building the dependency tree relation at a time.

While each action executes a special operation (such as creating a relation between tokens A and B, or changing the place of a token in the parser’s inner data structures), the role of the machine learning model is in each step to select the desired action out of all valid actions at that time. In graph-based parsing, the construction of the dependency tree is formulated by running e.g. the maximum spanning tree algorithm on top of fully connected graph with weighted relations between each token pair, and the role of the machine learning component is to give a probability weight for each relation in this fully connected graph.

### 1.3.6 Paraphrase modelling

Moving from structural analysis to a task more semantically oriented, in paraphrasing the same meaning is restated using different words. Paraphrasing occurs naturally in human communication, either by the same speaker repeating the message multiple times with different words, or multiple speakers conveying the same message in different places. While some of the paraphrases constitute only minor differences (e.g. one synonym replacement or structural modification), paraphrases can also use completely different structures and surface realizations without any lexical overlap in the statements. For example, the Finnish sentences

(a) *Olen siellä puolen tunnin päästä / I’ll be there in half an hour*

(b) *Saavun sinne 30 minuutin kuluttua / I will arrive in thirty minutes*

are paraphrases without any word-level lexical overlap. Common strategies to create paraphrases include e.g. synonym substitutions, negating antonyms, figurative language or metaphors, structural changes, and including redundancy or verbosity by including words not strictly necessary for the meaning (Bhagat and Hovy, 2013; Chang et al., 2021a). Few example of Finnish paraphrases including different modification strategies are shown in Table 1. In reality, typically paraphrasing does not invoke only one category of modifications but rather incorporates a mixture of different changes.

While a strict definition of paraphrases requires the two text statements to have exactly the same meaning, often in natural language processing and linguistic studies a broader definition is adopted requiring only having approximately the same meaning (Bhagat and Hovy, 2013). In NLP, paraphrasing leads to interesting challenges in different natural language understanding and generation tasks such as machine translation, machine reading, plagiarism detection, question answering and textual entailment (Mehdizadeh Seraj et al., 2015; Altheneyan and Menai, 2019; Soni and Roberts, 2019), each requiring deep understanding of the language. The NLP tasks directly modelling paraphrases are **paraphrase classification**, where given two text segments, the target is to determine whether the segments are paraphrases or not,

Statement 1	Statement 2
<b>Synonym replacement</b>	
Kissa on nopea. (The cat is fast.)	Kissa on vikkellä. (The cat is swift.)
<b>Negating antonym</b>	
Hän on elossa. (He is alive.)	Hän ei ole kuollut. (He is not dead.)
Hän ei ole täällä. (She is not here.)	Hän on jossain muualla. (She is elsewhere.)
<b>Metaphors</b>	
Olen täysin hereillä. (I am fully awake.)	Olen pirteä kuin peipponen. (I am bright-eyed and bushy-tailed.)
<b>Structural changes</b>	
Hän ei ole mikään kummajainen. (He is no weirdo.)	Ei hän mikään kummajainen ole. (Weirdo, that he is not.)
Kun hän lähti, hänellä oli...	Lähtiessään hänellä oli...
(When he left, he had...)	(On departure, he had...)
<b>Redundancy and verbosity</b>	
Hae pakkaus kaapista. (Grab the package from the cabinet.)	Hae pakkaus. Se on kaapissa. (Grab the package. It's in the cabinet.)

**Table 1.** Visualization of example changes to surface realization in order to create a paraphrase.

**paraphrase extraction or retrieval**, where given candidate documents or text corpus, the task is to extract text segments which are paraphrases with each other or find a paraphrase for the given text segment from the text collection, and **paraphrase generation or rephrasing**, where the objective is to rephrase the meaning of the given statement using different words. While paraphrase modelling has also direct applications (for example filtering duplicate questions from QA websites), many applications, e.g. in information retrieval or question answering systems, highly benefit from such semantic understanding, even if not directly applying paraphrase classification.

## 1.4 Corpus Annotation

Accessible text collections, called **text corpora**, are essential in NLP research, allowing us to study the language and learn its characteristics. While raw text corpora include only the original text with optional metadata information, **annotation** can be used to enrich the data with additional analysis or information. Annotated corpora are crucial for NLP research as they are used both in training supervised machine learning models as well as evaluating their performance.

When creating an annotated corpus, several aspects need to be considered. The **corpus size** tells how many examples in total are annotated, reflecting the purpose to which the corpus was created (e.g. training or evaluation). The **corpus domain** reflects the topics or text registers of the underlying texts, where a general domain corpus usually includes texts from different topics or registers and a domain-specific corpus targets into a special topic or register, e.g. such as medical or social media. The **annotation scheme** instructs and restricts the annotation work by defining what kind of labels are available and how each labels should be interpreted while annotating the examples. The annotation scheme also communicates the information related to the annotation between the corpus creators and the external users of the corpus. The annotation method can be either **manual or automatic annotation**, where in manual annotation a human assigns the labels for each example, while in automatic annotation (later referred to as automatic analysis in order to clearly separate the term from the manual annotation), the labels are assigned automatically by e.g. a machine learned model. While the automatic analysis is cheaper and faster than the manual annotation, and thus can be used to label larger text corpora, human annotation is almost always necessary in order to create high quality annotations. Sometimes annotation can also be obtained semi-automatically, where a human creates the annotation as a by-product of another process, creating for example metadata which can be automatically collected to serve as annotations for a specific task.

When creating a human annotated corpus, it is necessary to evaluate the reliability and consensus of the annotations. The quality of the annotations can be measured by using **double annotation**, where the same example is annotated individually by two or more different annotators, and the annotations are later compared, producing measures of **inter-annotator agreement (IAA)** evaluating the agreement of different annotators on the same data samples. In case of resolving the disagreements between different annotators and therefore producing a consolidated consensus subset of the annotations, the agreement can be measured between each annotator and the consensus subset using e.g. accuracy, the proportion of similarly annotated examples out of all examples, or using any metric suitable for the given task. The agreement between two annotators without a need for the resolved consensus subset, is typically measured using Cohen's Kappa (Cohen, 1960), which also takes into account the agreement happening by chance.

Next, we will describe two main resources enabling the work on Finnish syntactic parsing described in this thesis, the **Turku Dependency Treebank**, and its **Universal Dependencies** annotation scheme. I have contributed to these resources also outside the scope of this thesis by serving as an annotator during the original dependency annotation of the TDT corpus, participating in the design of the Finnish-specific UD guidelines (as part of the data conversion described later in this thesis), as well as maintaining the converted UD Finnish treebank in the UD collection during and after the thesis work. In the end of this section, we will also shortly introduce the

**Turku Paraphrase Corpus** created as part of the thesis work.

### 1.4.1 Turku Dependency Treebank

The Turku Dependency Treebank (TDT) by Haverinen et al. (2014) is a publicly available collection of manually annotated dependency trees of general Finnish including approximately 200,000 tokens (15,000 sentences) collected from 10 different text sources including multiple genres and topics. The text sources include Wikipedia articles, WikiNews, university online news, financial news, student magazine articles, blogs, fiction, Europarl speeches, JRC Acquis legislation, and grammar examples.

The main focus in the TDT corpus annotation was on dependency relations, including full manual double annotation for dependencies, where the double annotations were merged and all disagreements resolved collaboratively in order to produce the final consolidated consensus annotations. It is worth noting that I served as one of the annotators throughout the dependency annotation process, gaining a thorough understanding of the corpus and its annotation scheme. In addition to dependencies, also the text segmentation included manual revisions, whereas part-of-speech tags, morphological features and lemmas were automatically analysed in the original release. However, after the original release, manual verification of morphology and lemmas were included, making all layers of annotation fully manually annotated.

The dependency relations in TDT are based on the Stanford Dependencies (SD) annotation scheme (De Marneffe and Manning, 2008a,b) with few extensions to better suit the Finnish language, as the scheme was originally developed primarily on English. In addition to the *basic variant* of the SD annotation scheme, the TDT corpus introduced an extended dependency layer annotated on top of the base trees mostly adopted from those defined in the extended version of the SD annotation scheme, creating an enhanced graph analysis. The extended dependencies annotated in TDT include propagation of conjunct dependencies, external subjects, syntactic functions of relativizers, as well as gapping.

The annotation for part-of-speech tags, morphological features, and lemmas was carried out as single annotation using the OMorFi morphological transducer (Pirinen, 2008) as the initial starting point.<sup>1</sup> Each word was analysed using OMorFi obtaining from zero to several possible analyses each including a lemma, a part-of-speech tag, and a set of morphological features. In cases where one or more analyses were returned for a word, these analyses were manually checked and disambiguated based on the context. In case where none of the returned analyses were correct, or zero analysis was returned (unknown word for the transducer), the correct lemma and features were inserted manually.

---

<sup>1</sup>Note that at this point of work, the Universal Dependencies framework was not available yet.

In Haverinen et al. (2014), the dependency annotation accuracy of individual annotators is determined to be between 95.9–88.0<sup>2</sup> as measured between individual annotators against the final consensus annotations in terms of **labeled attachment score** (LAS, measuring the proportion of tokens with correctly assigned head and dependency relation in the base dependency tree). Furthermore, in triple annotation experiments, where a sample of double annotated sentences is yet annotated by a third expert annotator and the disagreements between this third annotator and the double annotated sample are settled to produce a super-gold sample, the double annotated data is shown to obtain 97.6% accuracy when measured against the super-gold sample, demonstrating the high quality of the treebank dependency annotations and giving an upper bound to measurable parser performance on the treebank.

## 1.4.2 Universal Dependencies

Universal Dependencies (UD) is a community lead effort to build cross-linguistically consistent treebank annotations for many typologically different languages (Nivre et al., 2016, 2020; de Marneffe et al., 2021). In its current state (version 2.12 (Zeman et al., 2023)) in addition to the annotation guidelines, the UD framework hosts community contributed treebanks for more than 140 languages, supporting comprehensive studies in multilingual parsing directly through different shared task (Zeman et al., 2017, 2018; Bouma et al., 2020a) as well as indirectly by providing unified resources. The cross-linguistically consistent UD annotation scheme is the de-facto standard scheme nowadays in multilingual parsing experiments supporting both linguistic and NLP studies across different languages.

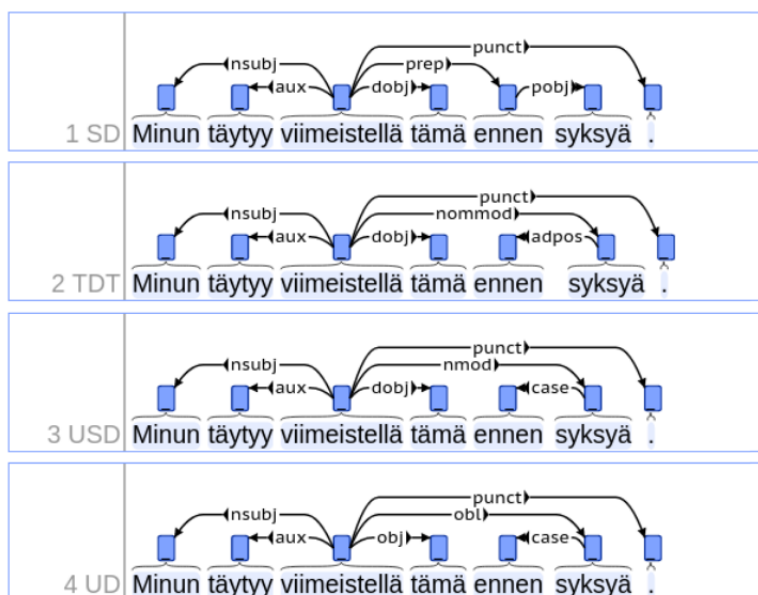
The dependency annotation in UD is revised based on the Stanford Dependencies scheme, especially its later variant of Universal Stanford Dependencies (De Marneffe et al., 2014). The morphological layer in UD on the other hand is based on the Google Universal part-of-speech (POS) tagset (Petrov et al., 2012) and the Intersect interlingua of morphosyntactic features (Zeman, 2008).

During the first UD data releases in 2015 the primary source of UD data was treebanks converted to UD from other dependency guidelines, later also natively annotated UD treebanks started to appear. The first release was based on data converted from the Google Universal Dependency Treebank Collection (McDonald et al., 2013), enhanced with several community donated treebanks from different research groups, also including our Turku Dependency Treebank. Therefore, the Turku Dependency Treebank was among the first datasets converted to UD, and released together with 9 other languages in the first UD release v1.0 (Nivre et al., 2015). The treebank conversion work is described as part of this thesis in Section 2.1 and in Paper I.

In Figure 2 we show a comparison of the original Stanford Dependencies (SD)

---

<sup>2</sup>Disregarding the least confident annotator, who annotated only 2.6% of the data with an agreement of 71.8.



**Figure 2.** Four related dependency annotation schemes in the order of their publication, (1) SD (2) TDT (modified SD) (3) Universal SD (4) UD (2.0). The English translation reads *I need\_to finish this before the \_autum.*

annotation scheme as defined for English, the SD scheme as modified for Finnish during the TDT annotation, the multilingual variant of the SD scheme (Universal SD) as well as the UD annotation scheme. Some of the language-specific adaptations presented in the TDT annotation for Finnish were later adopted by the multilingual SD scheme and therefore are also part of the current UD annotation scheme. The most notable such modification is the treatment of prepositional phrases and inflected nominal modifiers (concerning relations *prep*, *pobj*, *nommod*, *adpos*, *nmod*, *obl*, and *case* in the example). While the original SD scheme treats the preposition words the head of the prepositional phrase, despite using different relation names the rest considers the nominal modifier the head while the preposition word depends on the modifier.

### 1.4.3 Finnish Paraphrase Corpus

Turku Paraphrase Corpus, described in more detail in Section 3.1 and Paper V, is a manually annotated corpus of Finnish paraphrases including a total of 104,645 manually classified paraphrase pairs. The annotated paraphrases include pairs manually extracted from two related text documents with high probability of naturally occurring paraphrases (e.g. alternative translations of same source texts or different news articles describing the same event). Additionally, a small subset of the paraphrases

(12% of the data) are created through manually rewriting the original statements in order to create paraphrases equal in meaning in all reasonably imaginable contexts.

The paraphrases are collected from five distinct text sources including alternative movie and TV episode subtitles, news articles, discussion forum messages, university student translation exercises as well as university essays and exams. However, due to different factors related to annotation speed and data availability, most of the corpus data is obtained from the subtitling data. The corpus design is based on the principles of building a large but high quality corpus avoiding possible biases towards trivial, easily recognizable paraphrases often introduced with automatic candidate selection. The corpus annotation thus relies on manual candidate extraction, where the annotators manually select paraphrase pairs from two related documents, avoiding uninteresting pairs including only trivial differences easily recognizable e.g. with lexical overlap.

In addition to manual candidate extraction, all paraphrases in the corpus are also manually classified according to the annotation scheme developed together with the corpus. The classification scheme is developed for fine-grained paraphrase classification with four base labels: (4) universal paraphrase in all imaginable contexts, (3) paraphrase in the given context but not in all contexts, (2) related, but not a paraphrase, and (1) unrelated. In addition to base labels, several additional flags are applied to subcategorize the great amount of positive paraphrases which are not universal paraphrases for a specific reason, however, not entirely context dependent either.

## 1.5 Research Objectives and Disposition of the Thesis

The primary objective of the thesis is to advance Finnish natural language processing by providing appropriate resources and exploring best practises in their development and utilization. The thesis is structured into two distinct sections, namely syntactic parsing and paraphrase modelling, each with its own set of research questions.

When starting to work on this thesis, a reasonably sized, manually annotated treebank for Finnish already existed. However, when utilizing the treebank to train a machine learned parser for Finnish (Haverinen et al., 2014), the parsing numbers were only moderate compared with those published for some other languages. Therefore, the first section focuses on syntactic parsing and addresses the following research questions:

- (RQ1) Is Finnish inherently more challenging to parse with regards to accuracy when compared to other languages, such as English? Furthermore, given the existing Turku Dependency Treebank, how far can we advance in dependency parsing without the necessity to increase the size of the manually annotated corpus?**



- (RQ2) What methodological approaches should be employed to optimize the accuracy of the parsing pipeline?**
- (RQ3) Specifically focusing on lemmatization, what is the most effective approach to developing a machine-learned, context-aware lemmatizer, and how would its performance compare to hand-crafted grammatical rules?**

The contributions addressing these research questions are presented in Papers I–IV. Paper I (2015) outlines the conversion of the Turku Dependency Treebank into the Universal Dependencies annotation scheme, resulting in the creation of the UD Finnish-TDT corpus. This paper establishes the foundation for Finnish UD parsing and facilitates cross-language performance comparisons. The paper II (2020) introduces a novel lemmatization approach that achieves state-of-the-art performance across the UD treebanks. Alongside the development of the machine-learned lemmatizer model, the paper conducts an extensive data and system analysis. In Paper III (2018), the Turku Neural Parser Pipeline is detailed – a parsing system with the capacity to generate fully annotated dependency trees from raw text. In the CoNLL 2018 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies, this system ranked the 1st, 2nd and 2nd positions among 25 participants when evaluated across three different metrics, achieving a combined 1st place. This was the first paper to introduce the principle lemmatizer work later extended in the Paper II. Paper IV (2020) extends the pipeline’s capabilities by utilizing pre-trained, contextual language models rather than static word embeddings. This evolved version, the Turku Enhanced Parser Pipeline, ranked 1st among 10 participants in the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies.

The second part of the thesis shifts its focus to language understanding through an exploration of paraphrase modelling. With the significant advancements that the large, pre-trained language models have brought to different tasks — later demonstrated to include also syntactic parsing — the emphasis in the field has increasingly moved towards a deeper semantic understanding of language. The late success raises questions about whether the current models are genuinely comprehending language, rather than merely relying on simple surface cues. However, datasets designed to measure semantic comprehension in Finnish have been non-existent, or very scarce at the best.

Our primary research objective in paraphrase modelling centers around constructing a high-quality corpus of Finnish paraphrasing examples tailored for machine learning purposes. While the utilization of sentence-level heuristics is a prevalent approach for constructing large-scale paraphrase corpora, we hypothesize that these methods may introduce a bias towards shorter and simpler examples that can be automatically identified. Therefore, we pose the following research questions:

- (RQ4) Can the creation of a large-scale paraphrase corpus be efficiently accomplished by manually selecting examples, thereby mitigating bias towards**

	Paper I	Paper II	Paper III	Paper IV	Paper V
RQ1	✓		✓	✓	
RQ2		✓	✓	✓	
RQ3		✓	✓		
RQ4					✓
RQ5					✓

**Table 2.** The relation of research questions and papers in the thesis.

### shorter and simpler examples?

**(RQ5) Does the resulting corpus exhibit greater diversity in terms of example length and complexity compared to corpora where candidates are automatically generated?**

These questions are addressed in Paper V, where novel methodologies for paraphrase corpus creation are investigated. Within this paper, the Turku Paraphrase Corpus is introduced. It comprises 104,645 manually annotated Finnish paraphrase pairs, which are predominantly also manually selected from related documents pairs. Together with the dataset, the paper presents several corpus evaluation and comparison experiments as well as baseline results for different machine learned models trained and evaluated on the new data.

The relations of research questions and papers included in the thesis are summarized in Table 2. Papers II, III and IV present methodologies in a language-agnostic manner, reporting evaluation results across multiple languages and enabling comparisons between Finnish and other languages. In contrast, both Papers I and V focus specifically on Finnish, introducing new Finnish datasets and conducting Finnish-specific experiments. This thesis is based on co-authored publications, and while the experimentation process and manuscript preparation were shared efforts, distinct responsibilities were distributed among those contributing towards these papers. My individual contributions are outlined separately under the heading *Own contributions* in the beginning of the thesis.

## 2 From Raw Text to Dependency Graphs

In this chapter we address the first set of research questions relating to dependency parsing (RQ1, RQ2 and RQ3). As mentioned earlier in Section 1.4.1, the first, large-scale treebank for Finnish dependency parsing was released in 2014, creating necessary foundations for machine learning based parsing research for Finnish. However, the parsing results reported in the paper accompanying the data release demonstrated a clear gap in accuracy between automatic and human-made analysis (Haverinen et al., 2014). Furthermore, when compared to the accuracy of English parsers published roughly around the same time (e.g. Chen and Manning (2014)), the Finnish numbers clearly lag behind the English ones. This chapter summarizes the contributions of **Papers I, II, III and IV**, all of which are focused on narrowing down these performance gaps by first introducing a conversion of the treebank data into the Universal Dependencies scheme, and then building a state-of-the-art parsing pipeline for Finnish. While many of the technical details are not discussed here and can rather be found from the paper reprints, this chapter tries to highlight the most important contributions towards the presented research objectives, as well as give a broader overview of the research field in general, and therefore connect the individual papers to the related work on the field.

When summarizing the main work and results, special focus is given to evaluation and analysis in the Finnish language. However, the tools are designed to be multilingual, and in the original work are often evaluated in highly multilingual setting, enabling also cross-language comparison. Furthermore, many of the parsing results shown in this chapter are evaluated through several international shared tasks in the field focusing on multilingual parsing. While shared tasks have received also concerns related to good scientific practises (Parra Escartín et al., 2017), the overall objective is to advance research in the field by working together towards shared goals as well as giving a comprehensive and objective evaluation of different systems (Nissim et al., 2017).

Multilingual parsing evaluation has a strong backbone starting from 2006 through several shared tasks (Buchholz and Marsi, 2006; Nivre et al., 2007; Seddah et al., 2013) each including 9–13 languages on which the participant systems were evaluated in a controlled setting. In general, the shared tasks on multilingual parsing advance the inclusion of several languages not usually addressed in methodology oriented research papers, thus attracting wider audience to work on languages and

corpora outside the mainstream. The CoNLL 2017 and 2018 shared tasks on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al., 2017, 2018) greatly increased the number of evaluated languages including about 50 languages on each run of the shared task, as well as brought two important novelties to the scene; 1) cross-linguistically compatible Universal Dependencies scheme allowing better cross-language performance comparison, and 2) a tradition of evaluating parsing systems on top of predicted segmentation and morphological features instead of gold standard to give more realistic real-world evaluation scenario. These novelties created an excellent setting of evaluating strongly language-agnostic parsing pipelines.

When creating an overview of the results provided in this chapter, one must keep in mind that the different results published across the research timeline are difficult to directly compare due to the continuously improved conventions of UD specifications as well as improvements introduced to the actual data releases, therefore the underlying data undergoing some amount of changes between different UD versions. For the same reason, obtaining a detailed overview of the performance increase related to the introduced technical contributions may be difficult. However, in terms of our primary Finnish corpus used throughout the thesis, the modifications between different UD release versions are minor, and are not expected to greatly affect the comparison. The exception to this may be the UD release v2.0 including major structural changes both in the annotation guidelines and the actual annotations, as well as releasing a previously held-out test set for the public use. To account the comparability issues, one should only compare the general performance level without doing exact (decimal-level) comparison between systems trained on different versions of the data. However, when including comparative evaluation of systems using the same versions, e.g. all results from the same paper or same shared task, accurate and exact comparison can naturally be made.

## 2.1 Turku Dependency Treebank into Universal Dependencies

Data being annotated differently between different languages, or even between treebanks for the same language, may pose various issues in studies including multilingual or multitreebank aspects. For example, when evaluating the parsing performance, the expressiveness of the annotation scheme plays an important role. Naturally, the performance numbers of a parser using an annotation scheme with 50 relation labels cannot be directly compared to one including only 10 relation labels, and therefore when comparing the parsing performance across different languages it is important to account for these factors.<sup>1</sup> The same holds for many linguistic

---

<sup>1</sup>Of course, we note that the annotation scheme is not the only external factor affecting the parser performance, but the final performance is rather affected by a combination of several factors, for in-

studies where studying similar phenomena across languages becomes more difficult when expressing the analysis using different schemes. At the same time the different annotation policies between languages do not necessarily reflect these languages expressing certain phenomena with distinct structures but rather the convention of leaning into a diverse set of annotation conventions, thus sometimes receiving distinct analysis for identical structures between treebanks and languages. One of the design principles of the Universal Dependencies is to *facilitate consistent annotation of similar constructions across languages, while allowing language-specific extensions when necessary* (Nivre et al., 2016). With such consistent annotations the UD treebank collection better supports multilingual comparison of parsing performance, as well as supports multilingual studies e.g. in cross-lingual parsing and linguistics.

Taking this into consideration, the multilingual UD collection has a great potential of drawing interests towards studies involving several typologically different languages, or organizing shared tasks based on the large treebank collections. Therefore, including the Turku Dependency Treebank into such collection can be seen a major advantage greatly intensifying the attention towards Finnish parsing research, and already as such motivating to invest the time and effort needed for converting the original treebank annotations into the UD framework. From the perspective of our research questions, this multilingually consistent treebank annotation directly facilitates cross-lingual performance comparison. In this section, we summarize the process of converting the Turku Dependency Treebank into the Universal Dependencies annotation scheme from **Paper I**. After that, the main outcomes of the conversion work are discussed, further demonstrating the significance of the conversion work.

### 2.1.1 Overall Approach

As already discussed in Section 1.4.2, given the relatively similar annotation conventions used both in the original TDT and UD annotation guidelines, a relatively good conversion quality can be expected from automatic conversion techniques without major manual annotation effort. Therefore, the main conversion was implemented as a pipeline of automatic processing components designed separately for each annotation layer. The implementation is carried out by revising the conversion gradually based on manual inspection of the outcome. Manual adjustments were needed only for a relatively few cases where satisfying accuracy was not obtained using the automated pipeline, or investing time to automatically address only a handful of special cases was not feasible. While most of the conversion work took place before the first official UD release (v1.0 in 2015) and is therefore described in Paper I, the current version of the data have undergone several adjustments also after the initial work, including both minor corrections of errors in the data as well as necessary modifica-

---

stance treebank size or text genre.

tions caused by the evolving annotation guidelines.

### 2.1.2 Part-of-Speech and Morphological Features

The UD guidelines for part-of-speech tags defines 17 universal POS tags with a strict requirement of all treebanks conforming to use the same set without any language-specific additions. In turn, TDT uses 12 part-of-speech tags, however, some divided into several subcategories. In most cases, the division of words to POS categories is directly comparable between these two resources, therefore simple renaming of the tags is sufficient for most cases, potentially distinguishing on the subcategory level if needed. However, four part-of-speech tags used in the TDT annotation required distinction into several UD categories (namely pronoun into PRON or ADJ, punctuation into PUNCT or SYM, symbol into PUNCT or SYM, and verb into VERB and AUX), and different rules were written to handle these cases. The rules obtained the relevant information for instance from syntactic relations (e.g. the dependency relation `aux` indicating auxiliary usage of the verb), surface form -based heuristics (e.g. for distinguishing emoticons from other punctuation sequences), or wordlists (e.g. defining a list of Finnish proadjectives). In the conversion, two of the UD part-of-speech tags were left unused, DET for determiners and PART for particles. As the original TDT annotation does not distinguish between determiners and pronouns, or adverbs and particles, we opted not to introduce the distinction in the UD Finnish-TDT annotation either. For the secondary, language-specific part-of-speech tag annotation available in UD (XPOS), we simply use the original TDT tags as is.

For morphological features, a wide set of morphological categories including lexical or inflectional properties of words are defined in the UD annotation guidelines. Such categories include e.g. Case, Person, Number, Voice and Mood, each including its own set of possible values. Unlike for UD POS tags, in the case of morphological features the treebanks are allowed to introduce language-specific features not included in the universal feature definitions, thus the conversion process being more approving towards including the aspects of the original annotation. In order to minimize the information lost during the conversion, we decided to use the opportunity to introduce language-specific features when a corresponding feature was not defined in the universal feature set.<sup>2</sup> While most of the features could be directly mapped into UD with simple renaming, some required more complex methods, such as lemma lookup tables, especially when inserting new features not explicitly present in the original TDT annotation.

The part-of-speech and feature conversion was implemented as stand-alone scripts, in the input side reading TDT annotated words and producing UD annotated words as the output. In cases where the decision is context dependent and information out-

---

<sup>2</sup>Note that some of these features were included to the universal feature set later.

side the word itself is needed (e.g. its dependency relation), the conversion pipeline is implemented to introduce all possible UD alternatives for the input word, disambiguating these later in the pipeline when the relevant information becomes available. As will be described later in Section 2.2, this design allows us to utilize the same conversion tools during statistical parsing while still utilizing the OMorFI morphological analyzer as a preprocessing tool.

### 2.1.3 Lemmas

The initial lemma annotation in the first release was based on directly transferring manually annotated lemmas from TDT annotations to the UD version. However, the Finnish morphological analyzer OMorFI, which was used as a starting point in the manual TDT lemma annotation, included some discrepancies with UD guidelines, therefore several unfortunate systematic deviations from UD specifications were introduced during the first stage conversion. These were later addressed through several patterns of manual corrections, mostly focusing on issues related to derivational morphology and compounding.

While the specifications for lemma annotation in UD are quite loose and underspecified in many cases, UD instructs against normalizing derivational morphology, thus the English word *organizations* should be lemmatized as *organization*, not *(to) organize*. On the other hand, in TDT (due to the influence of OMorFI) derivational morphology was often normalized, leading to e.g. verb lemmas for noun derivations. For this reason, all words in the corpus including the morphological feature *Derivation* were manually inspected and necessary corrections made.

Another group of systematic lemma revisions relates to compounding words. While many of the noun compounds in their dictionary form are created by concatenating two noun lemmas (e.g. *ruoka* 'food' and *pöytä* 'table' — *ruokapöytä* 'dining table'), sometimes compounds are formed by connecting inflected, derived or shortened words as well (e.g. *oikeus* 'justice, court' in genitive case and *käynti* 'working, action' — *oikeudenkäynti* 'trial, legal proceedings') (Karlsson, 2015). In the latter case, the original TDT annotation often lemmatized both elements in the compounding word separately using a hash character to indicate internal word boundaries (e.g. *oikeus#käynti*). However, this created lemmas which as such do not appear in the language, even if not taking into account the internal boundary markers, and hence will not appear in the dictionary either, while UD suggests to lemmatize into "*canonical or base form of the word, which is the form typically found in dictionaries*"<sup>3</sup>. We decided to manually verify the lemma annotation of all compounding words, only normalizing inflections varying between different case variants of the compound. For example, *Uudessa-Seelannissa* 'New Zealand' in inessive case becomes *Uusi-*

<sup>3</sup>See documentation in <https://universaldependencies.org/u/overview/morphology.html>.

*Seelanti* where both parts are lemmatized, while *oikeudenkäynnissä* 'legal proceedings' in inessive case becomes *oikeuden#käynti*, lemmatizing only the second part of the compound. Also, we opted for keeping the internal boundary markers also in the UD version of the data, as these can, for instance, support information retrieval applications.

## 2.1.4 Dependency Relations

Typically, most time-consuming in treebank conversion are dependency relations as in addition to mapping the relation types to the new scheme, also structural reconfigurations are often needed. However, given the similarities between UD and original TDT relations, relatively lightweight and straightforward dependency conversion could be used, encountering fewer challenges than might be expected when converting from other annotation schemes.

In terms of relation labels, we were able to account for most of the dependency relations with automatic conversion rules. While most of the relation labels were either unmodified or required only a simple one-to-one label renaming, a handful of relation types required more complex one-to-many mapping where one TDT relation type was divided into several UD relation labels depending on the usage. For example, while most of the adverbial modifiers remained unchanged, TDT used this relation type also for sentence-initial conjunctions while UD annotated these with the relation meant for coordinate or subordinate conjunctions. Therefore, it was required to recognize all such cases, and rename the relation type accordingly.

Furthermore, two relation types were omitted without substitution as they were identified as enhanced relations that neither affect the base tree nor fall under UD enhanced relations, and a few relation types defined in the UD guidelines remained unused in the newly converted data. This occurred either because the grammatical relation is not traditionally used in Finnish (e.g. indirect object *iobj*), or because a comparable relation is not annotated in the TDT data (e.g. *list*). This lack of annotation makes automatic conversion unfeasible, and for relations where the expected frequency would be extremely scarce, we decided not to invest manual annotation effort to these. Finally, in several cases, the original TDT relation labeling included more detailed analysis as introduced in the universal label category of UD, and therefore we opted to include some of the original analyses by introducing language-specific relation subtypes for instance distinguishing between standard nominal subject and nominal copula subject (*nsubj* versus *nsubj:cop*).

Relatively few structural reconfigurations were required during the conversion, due to both SD and UD schemes already sharing most of the annotation principles by emphasizing direct relations between content words, while annotating functional words as dependents of these. In addition to the general scheme similarity, few modifications were made to the original SD scheme during the TDT annotation already



implemented some of the changes made in UD compared to the original SD, thus TDT directly implementing UD specifications regarding these structures. One such frequently appearing structure is prepositional phrases, where in TDT and UD the adposition word depends on the noun while SD (*basic* variant) treats the adposition as the head of the phrase. This, in fact, was in part due to the influence of TDT in the early stages of the UD scheme design.

The required structural changes included for example cases, where the TDT annotation allowed functional words to have dependents of their own creating a chained representation of e.g. auxiliary verbs while UD adopts a flat structure for these, as well as several head-final structures, where the final element of e.g. name or multi-word expression was considered head in TDT while UD adopts a head-initial structure. During the conversion, chains of functional words were reattached to the respective head words, and head-final structures were annotated as head-initial.<sup>4</sup>

The automatic dependency conversion was implemented as a rule-based tool searching for dependency relations matching given search criteria, and rewriting the relations to the ones defined in the given rule. The conversion tool was capable of both renaming existing relations under specific conditions, with rules such as *"search for a punct relation between tokens A and B, where the token B is an emoticon based on a given list of emoticons; rename the relation to be discourse"*, as well as structural reconfigurations by rules stating for example *"search for tokens A, B and C where there is an aux relation between tokens A and B, and B and C (chain of aux relations); replace the latter with a new aux relation from A to C"*. A total of 116 manually implemented rules were defined covering a great majority of the existing TDT relations. The remaining relations not covered by the rules (on the order of ~250 relations) were manually checked after the conversion, which was determined to be more efficient than trying to obtain an exhaustive automatic conversion of each and every relation.

The first stage conversion included relatively few manual revisions. The relations (and the data in general), however, required several rounds of quality control during the different UD releases. For example, during the UD release 2.0 the use of copula construction from equation and attribution was expanded to also cover e.g. location and possession, requiring major revision to the data. Part of the new revisions were done automatically, but many changes required also manual work, ensuring the quality of the output. This continuous work is necessary to maintain TDT in compliance with the updated UD guidelines. Due to its incremental nature distributed across time it is not formally described in any paper, nevertheless the maintenance of the resource should be considered as a contribution to the work presented in the thesis.

---

<sup>4</sup>Note that during the Paper I, names were kept as head-final, however, later on the head-initial structure of names was adopted to the Finnish-TDT corpus as well.

## 2.1.5 Discussion and Outcome

The first release of the Universal Dependencies treebanks (Nivre et al., 2015) in January 2015 included 10 languages: Czech, English, Finnish, French, German, Hungarian, Irish, Italian, Spanish and Swedish, the first Finnish treebank included in the UD collection being our conversion of the Turku Dependency Treebank.<sup>5</sup> By the release v2.12 (May 2023), the number of treebanks available through the UD collection has increased to 245, including 141 distinct languages. For Finnish, there are 4 distinct corpora available, of which three are provided by us and one by the University of Helsinki. In addition to our primary TDT corpus, we have published two additional evaluation datasets for Finnish: Finnish-PUD (Zeman et al., 2017) and Finnish-OOD (Kanerva and Ginter, 2022). The former is one of the 18 parallel test sets in which the same underlying text is translated into different languages, and therefore being optimal for cross-lingual comparisons. The latter, on the other hand, serves as an out-of-domain evaluation set with documents sampled from domains absent in the original treebank. I was the main annotator for both of these datasets and therefore these also contribute to the topic of this thesis, however, in order to maintain the scope, these will not be discussed in detail.

The data released through the UD collection has brought broader attention to our treebanks within the international parsing community. For example, our converted TDT treebank has been part of several shared tasks and system benchmarking data collections targeting into multilingual evaluation or cross-lingual parsing, see Zeman et al. (2017, 2018) and Bouma et al. (2020b, 2021) for shared tasks and Hu et al. (2020) for system benchmarking, thus gaining significant interest also in international research community.

Based on the treebank statistics presented by Nivre et al. (2016) (based on the UD release v1.2), most of the 37 treebanks included in the release are automatically converted with or without small amount of manual corrections. Based on the size of the original TDT treebank (15,000 sentences, 200,000 tokens), as well as a major part of the conversion effort falling into relatively monotonic work (such as renaming common relation types), full manual conversion was not considered a feasible option. Additionally, given the fact of highly related annotation conventions between the TDT and UD frameworks, we were able to obtain a high quality conversion by automating majority of the conversion pipeline, with only relatively few problematic cases resolved manually.

The quality of the treebank conversion is supported by the parsing results on the newly converted corpus described in Paper I (summarized also in the next section). These results show strong correspondence between the two treebank versions (treebank before and after the conversion), which supports the accuracy of the automatic

---

<sup>5</sup>During the first release, the corpus was named UD Finnish, however, later renamed into UD Finnish-TDT.

conversion process. While annotation decisions are shown to affect parsing performance, see e.g. Silveira and Manning (2015), both annotation schemes are roughly on same level of complexity by using similar amount of relation labels as well as implements the same structural principles of content-word headness. Therefore, the Labeled Attachment Score (LAS) of the parser trained before and after the conversion are expected to be roughly comparable, significant differences to any direction being likely due to problematic conversions. If the LAS is substantially higher after the conversion, the conversion may oversimplify the data by often falling into a default option, whereas substantial decrease in LAS may indicate unsystematic conversion policies introducing randomness to the data. The LAS of the parser pipeline trained on the original TDT corpus is 80.1, while the same parser pipeline trained on the converted treebank is 81.0, not showing a substantial change in either direction.

## 2.2 Background on Statistical Parsing of Finnish

In this section, we give the background of statistical parsing of Finnish and report the baseline UD parsing numbers from the first experiments conducted in Paper I. Also, related work for the early stages of neural UD Finnish parsing is discussed.

### 2.2.1 Pre-Neural Times

Upon the release of the original TDT corpus, Haverinen et al. (2014) released also the first openly available, statistical parsing pipeline for Finnish capable of analysing text given in its raw, unsegmented form into full morphosyntactic analysis including token and sentence segmentation, part-of-speech tags, morphological features, lemmas, and dependency trees. The pipeline was constructed from independently trained machine-learned components, where both the sentence splitting and tokenization components were implemented using corresponding modules from the Apache OpenNLP toolkit<sup>6</sup>, part-of-speech tags, morphological features and lemmas were predicted using the Conditional Random Fields (CRF) based Marmot tagger (Müller et al., 2013) utilizing pre-analyses of the Finnish two-level morphological analyzer OMorFi (Pirinen, 2008; Lindén et al., 2009), while the dependency trees were predicted using the Mate tools graph-based dependency parser (Bohnet, 2010). While the dependency parser and both token and sentence segmenters were trained straightforwardly using standard machine learning methods, in morphological tagging and lemmatization external lexical resources were used. As shown by Bohnet et al. (2013), a straightforward strategy of training a pure machine learned tagger did not yield top accuracy results for many morphologically rich languages, and thus a hybrid approach utilizing both lexical resources and machine learning was introduced

---

<sup>6</sup><http://opennlp.apache.org/>

in morphological tagging and lemmatization.

Finite state transducers (FSTs) implementing two-level morphology introduced by Koskenniemi (1984); Karttunen and Beesley (1992) are rule-based models encoding vocabulary and inflection rules for analyzing an inflected word into its lemma and morphological tags, giving a set of possible morphological readings (lemma, part-of-speech, and features) of every recognized word<sup>7</sup>. In this hybrid approach, the OMorFi morphological transducer was used to generate a set of possible readings for each word, which were subsequently introduced as features to the machine learned tagger, where two different constraints were tested. In *soft* constraint, the tagger was allowed to freely predict the output based on the given features, while in *hard* constraint the prediction was restricted to the given input readings, therefore the tagger effectively disambiguating the given readings in context. While the *soft* constraint showed better generalization across many morphologically rich languages in experiments conducted by Bohnet et al. (2013), for Finnish the *hard* constraint yielded the best overall result in part-of-speech and morphological tagging as well as in lemmatization, where the predicted lemma was obtained from the disambiguated reading together with the morphological tags. Therefore, the *hard* constraint was applied in the Finnish parsing pipeline. Words not included in the transducer lexicon did not receive any readings from the transducer, and therefore for these words the tagger predictions were used as is to obtain morphological features, while for lemma prediction the wordform was copied to the lemma field as is, in the absence of a better strategy at the time.

In Paper I, the original statistical parser pipeline was retrained with the new UD Finnish-TDT corpus, introducing the first parser for Finnish producing UD analysis. While the token and sentence segmenters as well as dependency parser were straightforwardly re-trained using the new data, morphological tagging and lemmatization needed an additional, on-the-fly conversion step in order to be able to utilize the OMorFi readings also while parsing into the UD framework. In this on-the-fly conversion, the OMorFi is first used to analyse all words in the input text, but before applying the machine learned disambiguation step, the morphology conversion script developed for the treebank conversion is applied to transform all morphological readings into UD. After the conversion, these readings can be introduced as features for the tagger.

The overall performance for the statistical parsing pipeline for Finnish as described above is shown in Table 3, where the performance is measured on the TDT corpus test set before and after the treebank conversion using the tagging constraint as originally optimized in Haverinen et al. (2014) (*Original pipeline*). The performance is reported on gold standard segmentation. When running this comparison

---

<sup>7</sup>The recognized word means the word belongs to a language as determined based on the given lexicon and inflection rules.

Treebank	POS	PM	FM	LAS	UAS
<i>Original pipeline</i>					
TDT (SD)	96.3	93.4	90.3	80.1	84.1
TDT (UD)	96.0	93.1	90.5	81.0	85.0
<i>Optimized pipeline</i>					
TDT (UD)	97.0	94.0	90.7	82.1	85.8

**Table 3.** Results of the first UD Finnish parsing experiments using gold standard segmentation. TDT (SD) refers to the morphological tagset and dependency relations as defined in the original TDT after manual morphology annotation, and TDT (UD) refers to the UD Finnish-TDT data as released in UD v1.0. POS is the POS tagging accuracy, PM the accuracy of part-of-speech tag and all morphological features, FM is the accuracy of full morphology (part-of-speech, features and lemma).

with identical experimental setting before and after the data conversion, the evaluation results show strong correspondence between the two treebank versions, obtaining LAS 80.1 on data before the conversion and LAS 81.0 after it, supporting the high quality of our automatic conversion process. Additionally, when optimizing the tagging constraint on UD converted data (*Optimized pipeline*), the best Labeled Attachment Score (LAS) of the parser pipeline obtained after the UD conversion is 82.1, serving as a baseline for later studies involving UD Finnish parsing.

## 2.2.2 Early Steps in Neural Parsing of Finnish

UDPipe (Straka et al., 2016) was one of the first neural parsers with published results for the UD Finnish-TDT treebank in 2016, and afterwards it has served as the de-facto baseline for UD parsing for many years. UDPipe implemented the full pipeline of tasks supported by the UD annotation, including word and sentence segmentation, part-of-speech and morphological tagging, lemmatization and dependency parsing, excluding only enhanced dependency relations. The parser component in UDPipe, Parsito (Straka et al., 2015), incorporates distributed representations of input elements (words, part-of-speech tags, and morphological features) thus being able to utilize pre-trained word embeddings learned using for example word2vec or similar methods. This allows for utilizing large-scale, unlabeled resources in order to learn good representations of words’ meanings and usage prior the actual parser training.

When comparing the initial numbers released from the UDPipe system to our own Finnish statistical parsing pipeline described in the previous section, the LAS performance of UDPipe was estimated to be about 5pp lower (LAS 82 compared to 77) when evaluating both on top of gold segmentation. However, note that the test set used here is slightly different. While our pipeline was evaluated on the secret, held-out test portion of the corpus, the UDPipe was evaluated on the public test portion, and while the data characteristics is not expected to change between

the sets, the underlying test data is different, thus not giving exact comparison between the two systems.<sup>8</sup> Furthermore, our pipeline incorporated language-specific information (morphological transducer) not available if restricting the system to be language-agnostic as is the design principle of the UDPipe system. Additionally, the UDPipe did not utilize the full power of pre-trained word embeddings as it was using embeddings trained on the treebank training data only.

The CoNLL 2017 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies was the first parsing shared task, which included a Finnish treebank. The shared task was highly multilingual in its nature, especially calling for systems able to obtain high performance in language-agnostic manner, where the system does not require much or any language-specific adaptation. In total out of the 64 treebanks involved in the shared task, three were Finnish; UD Finnish-TDT, UD Finnish-FTB<sup>9</sup> developed at the University of Helsinki, as well as the UD Finnish-PUD annotated by us as part of the shared task organization serving as one of the parallel test sets across 14 languages, where each parallel treebank includes the same underlying text translated into different languages, therefore serving as an optimal material for cross-lingual parser comparison.

The winner of the CoNLL 2017 Shared task was the Stanford team (Dozat et al., 2017) utilizing their freshly introduced deep biaffine parser (Dozat and Manning, 2017), a graph based approach utilizing representations induced using recurrent encoder. The performance of the Stanford system on UD Finnish-TDT corpus was 85.6% on top of predicted segmentation, clearly outperforming the previous state-of-the-art. While the outcomes of the CoNLL-17 shared task showed positive impact to Finnish parsing performance, we were not satisfied to the current lemmatization performance, the best performing system yielding only an accuracy of 86.5 on the UD Finnish-TDT corpus (Che et al., 2017; Yu et al., 2017). Therefore, our next effort focused on improving the lemmatizer available for Finnish.

## 2.3 Sequence-to-Sequence Lemmatizer for Finnish

While our primary motivation was to develop a state-of-the-art lemmatizer specifically for Finnish, we designed the model to be language-agnostic, ensuring its applicability to all UD languages. The novel lemmatizer developed as part of the thesis work, Universal Lemmatizer, is chronologically introduced first time in Paper III, however, the Paper II includes an extended analysis and additional experiments in the area of lemmatizing UD data. The lemmatization work is summarized in this section.

---

<sup>8</sup>In the UD v2.0 release (during the CoNLL 2017 Shared Task), the earlier public test set was merged with the development set, while the secret test set was released as the new official test set.

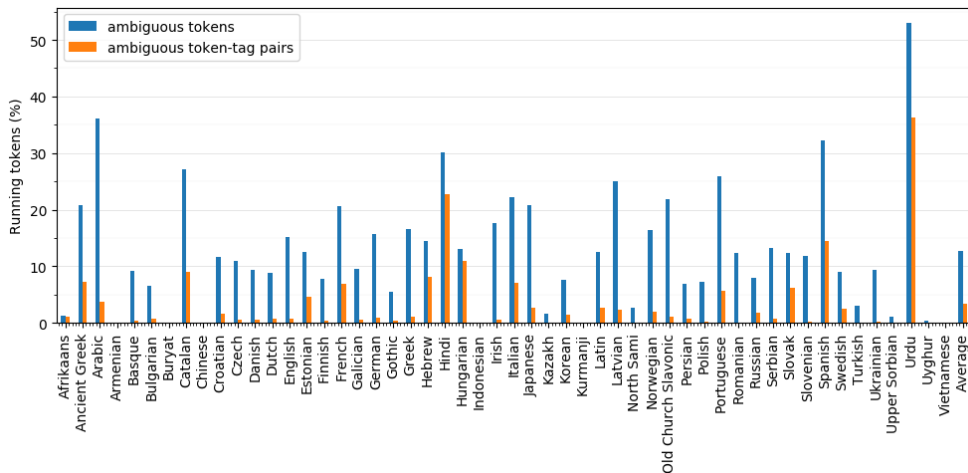
<sup>9</sup>[https://github.com/UniversalDependencies/UD\\_Finnish-FTB](https://github.com/UniversalDependencies/UD_Finnish-FTB)

### 2.3.1 On the Contextuality of Lemmatization

Most inflected words can be unambiguously lemmatized based on the word itself without requiring surrounding context, as they can be meaningfully derived from only one possible lemma (e.g. the Finnish inflected word *talossa* 'in a house' is generally recognized to have only one plausible lemma *talo*). However, every language contains instances of ambiguous inflections, where a single inflected form can originate from two or more distinct lemmas, necessitating contextual information to determine the correct lemma. The same holds also for out-of-vocabulary words, where context plays a crucial role in lemmatizing words seen for the first time. This means that by seeing only the inflected word, one cannot reliably determine its lemma without information about the context. One example of an inflected, Finnish word being ambiguous in terms of lemmatization is *koirasta*, which can be inflected from two distinct lemmas, *koira* 'a dog' or *koiras* 'a male'. However, note that the lemma ambiguity must not be confused with word-sense ambiguity where a word may be ambiguous in terms of its meaning while still having exactly one plausible lemma analysis. As an example, the Finnish word *kurkku* has two distinct meanings, *a throat* and *a cucumber*, however, both meanings share exactly the same inflection paradigm, where all meaning-ambiguous inflections are lemmatized into the common lemma, thus the word being unambiguous in terms of lemmatization.

The desired lemma for ambiguous words can be inferred based on the structure or meaning of the sentence. While in the *koirasta* 'a dog/male' example the ambiguous inflected word appears in different position/function in the sentence depending on the lemma, and thus the desired lemma can be correctly inferred based on morphosyntactic properties of the word in the known context. However, some other ambiguous words may appear in similar functions, in which case an understanding of the word's meaning in the context is required. In order to estimate the number of lemma-ambiguous words in different UD treebanks, as well as the number of cases where the known structural information is enough for inferring the correct lemma for an inflected word, we measure the percentage of running tokens with ambiguous lemma in the UD treebanks. For each language we measure two categories of ambiguities, firstly, the number of inflected tokens having more than one distinct lemma occurring in the treebank annotations, and secondly, the number of inflected tokens together with their morphosyntactic analysis (part-of-speech and morphological features) having more than one distinct lemma occurring in the treebank annotations. While the first measures the percentage of plain lemma ambiguity occurring in the treebanks, the second measures the percentage of tokens where the sentence structure itself is not enough for unambiguously inferring the desired lemma for the token.

The measures are shown in Figure 3. While the plain lemma ambiguity is close to 12% on average, the proportion of ambiguous lemmas drastically drops for most languages when the token's morphosyntactic tags are taken into account, the token-



**Figure 3.** Percentage of running tokens with ambiguous lemma and token-tag pairs with ambiguous lemma calculated from the UD v2.2 training data. All treebanks of one language are pooled together. [This figure was originally published in Paper II.]

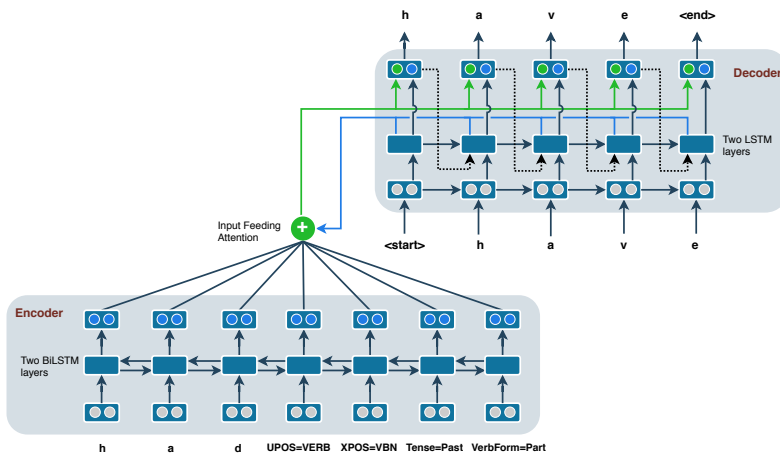
tag pair ambiguity being only close to 3% of running tokens on average. Additionally, for more than half of the languages, the ambiguity drops below 1% of running tokens when including morphosyntactic tags, demonstrating that the tags are a powerful yet compact contextualized feature representation for lemmatization ambiguity. For Finnish, while the number of ambiguous tokens is about 7%, the number of ambiguous (token, tag) -pairs is only ~0.5%, which translates into a very high oracle accuracy (upper bound) on Finnish when building a lemmatization system utilizing only morphosyntactic features in contextual lemmatization, at the same time giving a compact contextual representation compared to the full sentential context.

Based on this observation, we set out to build a sequence-to-sequence lemmatizer directly utilizing morphosyntactic information instead of representing the whole surrounding sentence as features.

### 2.3.2 Modelling Lemmatization Using Sequence-to-Sequence Framework

The recent neural lemmatizers typically fall into two different machine learning frameworks: sequence-to-sequence generation models (Bergmanis and Goldwater, 2018; Qi et al., 2018) or classifiers predicting edit-tree transformation rules (Müller et al., 2015; Chakrabarty et al., 2017; Straka, 2018), both frameworks being able to perform the lemmatization with or without contextual information. We select the generation approach for its flexibility and cast lemmatization as a sequence-to-sequence rewrite problem where lemma characters are generated one at a time from





**Figure 4.** Neural lemmatizer model implemented as sequence-to-sequence transformation. [This figure was originally published in Paper II.]

the given sequence of word characters and the word’s morphosyntactic tags. Once cast in this manner, essentially any of the sequence-to-sequence model architectures can be applied to the problem.

While sequence-to-sequence learning framework includes a great variety of different applications and has lately become an intensively researched topic in the field, maybe one of the most popular applications falling under the framework is machine translation including large and active community with many mature model implementation libraries. Therefore, in this work we rely on an existing neural machine translation model implementation by Klein et al. (2017). The model architecture together with an example input–output sequence, is illustrated in Figure 4. The items in the input sequence (individual characters as well as individual morphosyntactic tags) are encoded using two bidirectional LSTM layers to create contextual representation of each item in the input, while the output sequence is generated by a decoder with two unidirectional LSTM layers with attention over the encoder representation. While the encoder-decoder model is build using recurrent layers, there is no limitations towards specific layer structures, thus replacing the recurrent layers with e.g. self-attention layers (Vaswani et al., 2017) is entirely possible. However, during preliminary studies, an indication towards self-attention being more beneficial was not noticed.

Using this framework, the lemmatizer models can be trained by creating training examples using individual tokens from the UD treebanks (or any other resources including tokens with their morphosyntactic analysis). During training time, gold standard tags are used, however, at prediction time the evaluation can be based on gold standard tags or predicted morphosyntactic tags in order to get more reliable real-life evaluation numbers.

### 2.3.3 Results and Discussion

The evaluation is carried out on 76 treebanks representing 52 different languages from the UD v2.2. release. Primarily, we measure macro-accuracy over all evaluation treebanks, however, we also measure the macro-accuracy metric separately for different treebank groups as determined based on the amount of training data available (big, small and low-resource), or the design principle (PUD, which includes treebanks with the same underlying text translated into several different languages). The evaluation setting is adopted from the CoNLL 2018 shared task. The macro-average accuracy of our lemmatizer is above 92% when measured across all treebanks using predicted segmentation and predicted morphosyntactic features. When using the same task setting, but excluding treebanks from the low-resource category (5 treebanks, each having only close to 20 sentences in the training sets), the macro-average accuracy is above 95%, with individual treebanks ranging between 82.9% (Hebrew-HTB) and 99.7% (Indonesian-GSD). For UD Finnish-TDT, the lemmatization performance is 95.40 on predicted segmentation and morphosyntactic features, improving more than 8pp absolute compared to the CoNLL-17 level.

When comparing our system with several recent baselines used in Paper II, UDPipe v1.2 (Straka and Straková, 2017), UDPipe Future (Straka, 2018), Stanford (Qi et al., 2018), Lematus (Bergmanis and Goldwater, 2018), and dictionary lookup, our lemmatizer outperforms the other tested systems in three treebank categories (big, PUD, and small) with approximately 1.4-8.5 absolute points. While both UDPipe v1.2 and UDPipe Future are based on the edit-tree classification, Stanford and Lematus apply sequence-to-sequence approach. The Stanford system uses dictionary lookup for words seen in the training data while training a context-free sequence-to-sequence backup model for unknown words. Lematus is a context-aware sequence-to-sequence lemmatizer, where the word is given together with 20 characters from its immediate context on both sides of the word. The above mentioned evaluation results show that our lemmatization approach outperforms the baselines in cases where reasonable amount of training data is available. We see the advantage of our model being the combination of flexibility of the sequence-to-sequence approach combined with the easily learnable, dense contextual representation. In the fourth treebank category, low-resource, where it's apparent a good neural lemmatizer cannot be trained using only 20 training sentences without any kind of pre-training of the model, the simplest of the machine learned baseline models (UDPipe v1.2, where the lemmatizer is in practice limited to select one of the very few possible edit-trees seen in the training data) gives the best results. However, the naive dictionary lookup over training data performs comparably to the other systems in this category, the overall macro-accuracy in low-resource setting being between 40–65% for all systems, making all of these impractical for real applications.

In Paper II we also study several data augmentation techniques. Especially in ex-

tremely low-resource settings, where the off-the-shelf model performance is shown to be weak, model pre-training, training data augmentation or cross-lingual transfer methods could be potential for lifting the model performance. One additional advantage of our compact context representation is the possibility to utilize (word, tags, lemma) -triplets isolated from their natural sentence context. This allows us to rather straightforwardly use manually annotated resources through related projects including such information, for instance data extracted from Wiktionary<sup>10</sup> (Ylönen, 2022) or rule-based morphological transducers, to generate additional training examples for our system.

In Paper II we conduct a study where we lean on the Apertium morphological transducers capable of returning all known (tags, lemma) -tuples for given input words. While the morphological transducer itself does not disambiguate the possible readings in context, the transducer output can easily be used to create individual examples for our training regime without any disambiguation, only needing a language-agnostic feature mapping from Apertium morphological features into UD. When evaluating this data augmentation approach on four of the five low-resource languages (Armenian, Buryat, Kazakh, and Kurmanji, which include a morphological transducer in the Apertium), we notice extremely good generalization performance if reliable morphosyntactic features are available. When measuring the lemmatization accuracy on gold standard morphosyntactic tags, the lemmatization performance is 91%–96%. However, if using predicted tags from a tagger trained on the available treebank data (~20 training sentences per language), the lemmatization accuracy drops to 58%–74% hinting on severe error propagation. Therefore it is still an open question how to obtain reliable tagger predictions in order to utilize the potential of our data augmented lemmatization models for these low-resource languages.

## 2.4 Turku Neural Parser Pipeline

During the lemmatization work, we wanted to evaluate our lemmatizer models as part of the CoNLL 2018 shared task on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al., 2018) in order to obtain comprehensive comparison with lemmatization approaches used in current state-of-the-art parsing systems. To this end, we needed to build a full pipeline capable of carrying out segmentation, morphological tagging, parsing, and lemmatization steps for given raw text input. In addition to participating in the CoNLL 2018 shared task, we also wanted to provide an upgraded version of the Finnish parser pipeline to support state-of-the-art structural analysis of Finnish language.

To this end, we developed Turku Neural Parser Pipeline introduced in **Paper**

---

<sup>10</sup><https://wiktionary.org>

**III.** Our overall objective was to develop an easy-to-use parsing pipeline to be used for analysing raw text for different downstream applications. Although our main interest was in Finnish language, we also strove for the pipeline to perform well on other languages and all treebank sizes without requiring any language or treebank specific method adaptation. With this in mind and based on the knowledge learned from the CoNLL 2017 shared task results, we decided to rely on openly available, then state-of-the-art components for segmentation, tagging and parsing, adapted to our purposes when necessary, while integrating our own lemmatizer introduced in the previous section.

Next we summarize the initial Turku neural parser pipeline as it appeared in the CoNLL 2018 shared task submission and is described in Paper III. While this section describes the parser as introduced originally, the upcoming Sections 2.5 and 2.6 describe the improvements done after the shared task, as well as the current state-of-the-art in Finnish parsing.

### 2.4.1 Parser Modules

**Segmentation** The segmenter module of the pipeline is based on UDPipe by Straka and Straková (2017), which jointly performs sentence and token boundary detection using single layer bidirectional GRU network. For each character, the network predicts whether the character is the last one in a sentence marking both sentence and token boundary, the last one in a token marking token boundary, or no segmentation boundary character. After boundary detection the UDPipe segmenter includes a separate model for multiword token (MWT) expansion in order to split tokens such as *ettei* in Finnish (see Section 1.3.1 for more detailed explanation of MWTs). The MWT expansion model generates a set of suffix rules learned from the training data to split and expand multiword tokens into two syntactic words.

**Part-of-Speech and Morphological Tagging** The part-of-speech and morphological tagging module is an adaption of Dozat et al. (2017). The tagger is a time-distributed affine classifier over tokens in a sentence including two classification layers, one meant for universal part-of-speech tags and one for language-specific part-of-speech tags in UD. While the original tagger was not designed to predict the morphological features, in our modification we included the prediction of morphological features by simply concatenating the language-specific POS tag and all morphological features into a single prediction task by predicting the full concatenated string, such as *NOUN/Case=Nom/Number=Sing* as one class in multiclass classification. Therefore, the internal structure of the original model was not changed but rather we adapted the label set used in the multiclass classification. Both classification layers, UPOS and combined XPOS plus FEATS, utilize shared token embeddings contextualized using a bidirectional LSTM layer. These token embeddings are a combination

of pre-trained word embeddings, randomly initialized word embeddings, and a representation built from sequences of characters created with an unidirectional LSTM layer.

**Lemmatization** The lemmatization component of the pipeline is the Universal Lemmatizer from the Paper II. In the pipeline, the lemmatizer component is positioned after the tagger, and therefore it is able to utilize the part-of-speech and morphological tags predicted by the tagger. Together with the sequence-to-sequence lemmatizer, a pre-computed lemma cache is used to prevent computing the lemma multiple times for same input features. Each time the lemmatizer is launched, the lemma cache is prefilled with examples from the training data, and dynamically extended with all lemmatized words so that the output is computed only once for each unique input features during each running instance of the lemmatizer. For models where the contextual information is represented using the nearby lexical context, this approach would not be practical due to data sparsity, however, the dense context representation of the Universal Lemmatizer makes such cache feasible to maintain.

**Dependency Parsing** The dependency parser used in the pipeline is the graph-based parser by Dozat et al. (2017). In this parser, each token is first embedded as a combination of pre-trained, randomly initialized, and character LSTM -based token embeddings together with embeddings for the part-of-speech tags. These token representations are contextualized using bidirectional LSTM layers, obtaining the contextualized representation for each token from the final LSTM layer. Token representations are then projected using four different ReLU layers to two, jointly trained biaffine classifiers, one for deciding a head for each token by computing a score for each token pair, and one for deciding a label for each dependent-head pair by computing a score for each label for the given token pair. Our module follows the original implementation of the parser, the only difference being in the parser input where also morphological features are utilized. Since our part-of-speech tagger was modified to predict morphological features together with the language-specific part-of-speech tag as single classification, as a consequence the parser obtains information about the predicted morphological features through the input embeddings originally embedding language-specific part-of-speech tags, but in our pipeline including a single embedding for a combination of the language-specific part-of-speech tag and morphological features.

**Pipeline Structure** The parsing pipeline is designed to produce full morphosyntactic analysis from raw text into dependency trees using a single command, therefore making the usage easy for people outside the core NLP community as well. The pipeline is a combination of the above mentioned modules with a possibility for

easily including additional modules for optional text processing (e.g. cleaning noisy Internet language), or dropping some of the existing modules if not needed at the moment. The pipeline runs the defined modules in the given order, as it's common to have another module as prerequisite. The modules are run parallel trying to avoid stalling, where the next module would be waiting for the former to finish. Therefore, the input is divided into smaller batches, where each batch is forwarded into the next module immediately after the current processing is finished, while the current module starts to process the next batch simultaneously.

## 2.4.2 Turku Neural Parser Pipeline at CoNLL-18 Shared Task

As mentioned previously, the CoNLL 2018 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies continued the tradition of CoNLL 2017 shared task by evaluating the participating systems on a highly multilingual treebank collection using predicted segmentation as well as encouraging participant to provide also intermediate analysis such as part-of-speech tags and lemmas. While the primary metric in the CoNLL 2017 shared task was the widely used labeled attachment score (LAS) involving only dependency relations, the CoNLL 2018 shared task introduced three primary metrics, LAS, morphology-aware LAS (MLAS, the proportion of tokens with correct head, a subset of morphological features and functional dependents correctly predicted), and bi-lexical dependency score (BLEX, the proportion of head-dependent content word pairs whose dependency relation and both lemmas are correct), where MLAS and BLEX involved the evaluation of morphological features or lemmas together with the dependency relations. This further encouraged participants for working on systems with accurate prediction of intermediate steps as well, partly impacting on our decision of participating the shared task with our lemmatization work.

The Turku neural parser pipeline ranked 1st (BLEX), 2nd (MLAS) and 2nd (LAS) on the three primary evaluation metrics in the shared task among 25 participants. The results are summarized in Table 4 for all different metrics comparing our system with the best competitor in each metric in terms of macro average over all shared task treebanks. Together with this comparison, the official rank of our system is given for all different evaluation metrics. Reflecting the design principles of our pipeline, we ranked 1st on the two metrics evaluating lemmatization, *Lemmas* and *BLEX*, while the system ranked in the top 5 on all metrics.

In terms of Finnish, our pipeline obtained LAS score above 86%, UPOS accuracy above 96%, and lemmatization accuracy above 95% on the UD Finnish-TDT treebank. As a ballpark comparison, the original Finnish parser pipeline from Haverinen et al. (2014) had approximately 5pp worse LAS score, 2pp worse part-of-speech score, and more than 3pp worse lemmatization score. One must keep in mind here that the numbers are not directly comparable, however, the difference is still indica-

<b>Metric</b>	<b>Ours</b>	<b>Rank</b>	<b>Best Competitor</b>
Tokens	97.83	4.	98.42 (Che et al., 2018b)
Words	97.42	5.	98.18 (Smith et al., 2018)
Sentences	83.03	5.	83.87 (Che et al., 2018b)
UPOS	89.81	4.	90.91 (Smith et al., 2018)
XPOS	86.17	3.	86.67 (Straka, 2018)
Features	86.70	3.	87.59 (Smith et al., 2018)
All Tags	79.83	2.	80.30 (Straka, 2018)
Lemmas	<b>91.24</b>	<b>1.</b>	89.32 (Straka, 2018)
UAS	77.97	4.	80.51 (Che et al., 2018b)
CLAS	69.40	2.	72.36 (Che et al., 2018b)
<b>Primary Metrics</b>			
LAS	73.28	2.	75.84 (Che et al., 2018b)
MLAS	60.99	2.	61.25 (Straka, 2018)
BLEX	<b>66.09</b>	<b>1.</b>	65.33 (Che et al., 2018b)

**Table 4.** CoNLL 2018 official results for different metrics when measuring macro average over all shared task treebanks.

tive of the overall direction. It’s evident that the two CoNLL shared tasks affected positively to the accuracy of Finnish parsing.

## 2.5 Turku Enhanced Parser Pipeline

Paper IV continues the work on the Turku neural parser pipeline, and extends the pipeline in two directions: 1) utilizing pre-trained Bidirectional Encoder Representations from Transformers -model (BERT) as contextualized representation of the input words in a sentence rather than word embeddings, and 2) producing dependency graphs with enhanced UD relations instead of dependency trees. Otherwise, the pipeline adheres to the design principles of the Turku neural parser pipeline, leaving the segmentation and lemmatization components untouched, while upgrading the tagging and parsing modules.

### 2.5.1 Pre-Trained Contextualized Language Models

Transfer learning using the pre-train–fine-tune paradigm of massive language models have shown its potential across different NLP tasks (Ruder et al., 2019). In this paradigm the language modelling objective is used to pre-train the feature representation on unannotated corpora in order to obtain a general representation of the language, while later fine-tuning the learned representation into the specific task by using task specific, supervised training data. Especially with the existence of contex-

tualized language models, such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), GPT (Radford et al., 2019), or T5 (Raffel et al., 2020), capable of producing contextualized word embeddings and/or embeddings for text segments longer than words, contrasting to earlier distributional semantics approaches such as word embeddings (like word2vec and GloVe), numerous NLP tasks requiring deep language understanding have recently gained promising results. For example, in different natural language understanding benchmarks such models have helped to substantially narrow down the gap between human and model performance (see e.g. Wang et al. (2018), Wang et al. (2019), Raffel et al. (2020), Sun et al. (2021)).

The contextualized language models have been widely applied in dependency parsing as well. Che et al. (2018a) modified the Dozat et al. (2017) biaffine parser by replacing the embedding layers with deep contextualized word embeddings (ELMo) pre-trained on large web crawl corpora. Kulmizev et al. (2019) on the other hand experimented with both transition-based and graph-based dependency parsers utilizing ELMo or BERT pre-trained language models. Both of these studies utilized monolingual fine-tuning on top of either monolingual ELMo models, or the Google’s multilingual BERT model (mBERT). Kondratyuk and Straka (2019) presented a multilingual multi-task model referred to as Udify, where the mBERT model was fine-tuned into 75 languages simultaneously, thus producing a single model capable of creating predictions for 75 different languages, and four layers of morphosyntactic analysis (part-of-speech and morphological tagging, lemmatization, and dependency parsing). While the Udify model showed state-of-the-art performance for many languages, its accuracy for Finnish was clearly below the baseline performance. As we later showed in Virtanen et al. (2019), the Udify model architecture itself is perfectly suitable for producing state-of-the-art results also for Finnish. In the original study there were two independent reasons for the failure in terms of Finnish parsing accuracy. Firstly, in the study the Udify model was trained simultaneously for 75 UD languages using all available training resources for each language, therefore meaning the two independent Finnish treebanks including training data, UD Finnish-TDT and UD Finnish-FTB, were merged during training, and therefore the model learned from the mixture of the Finnish training data. However, previous studies have shown that both cross-treebank experiments as well as combining the training sets of the two Finnish treebanks will cause substantial decrease in parsing performance (Aufrant et al., 2017; Vilares and Gómez-Rodríguez, 2017), the likely cause being annotation inconsistencies between the treebanks. Secondly, the original study trained the Udify model by fine-tuning the multilingual mBERT model, which was later shown to be sub-optimal for Finnish.

However, based on the general success of incorporating pre-trained language models for dependency parsing, especially the Udify’s multitask fine-tuning approach showing state-of-the-art results for many languages, in Virtanen et al. (2019) we fine-tuned the Udify multitask network separately for each UD Finnish treebank using the



language-specific FinBERT language model<sup>11</sup> as the starting point. The study led to a new state-of-the-art performance on Finnish parsing, outperforming strong baselines by more than 3pp in LAS (absolute). Based on this positive outcome, the Turku neural parser pipeline was updated to support pre-trained BERT models using the Udify multitask predictor in Paper IV.

Furthermore, following the positive results of studies introducing other language-specific BERT models (e.g. de Vries et al. (2019), Kuratov and Arhipov (2019), Martin et al. (2020)), we hypothesized the parsing performance can substantially improve for several languages with dedicated language-specific models compared to the mBERT model, the de-facto standard multilingual BERT model at that time. In particular, in Virtanen et al. (2019) we report a +4.95pp absolute improvement in LAS when switching from the mBERT language model to the FinBERT model, when fine-tuning the Udify parser on the UD Finnish-TDT corpus. To follow this, in Paper IV we carried out a systematic study of language-specific (or language-group-specific) BERT models compared with the mBERT model, training new BERT models based on Wikipedia dataset (WikiBERTs, Pyysalo et al. (2021)) especially for languages where language-specific BERT models did not exist beforehand.

Out of 21 treebanks (17 languages) we found the language-specific BERT model (either an existing language-specific model or our newly trained WikiBERT model) outperform mBERT on 18 treebanks, when fine-tuning the Udify parser for each treebank separately, and measured on the development section in terms of LAS. However, in most of the cases, the difference is only moderate, the average improvement being 1.5pp (reducing errors an approx. 13%) over all treebanks. When comparing different treebanks and languages, the largest improvement is in fact seen for Finnish (+5.0pp), followed by Swedish (+3.6pp) and Lithuanian (+2.2pp).

## 2.5.2 Enhanced Graph Representation

Many downstream applications capable of utilizing syntactic analysis, such as open-domain relation extraction or biomedical event extraction benefit from two words or entities being directly connected in the syntactic analysis (Bassignana et al., 2023). While the UD representation is generally well suited for such downstream applications by directly connecting content words rather than connecting them through functional words, many phenomena remain hard to capture through single relations between words. For such reasons, many syntactic annotation schemes define also additional annotation layers suitable for annotating structures beyond syntactic trees, producing a graph output. As discussed already in Sections 1.4.1 and 1.4.2, the Stanford Dependencies annotation scheme, used in the annotation of Turku Dependency Treebank, defines an extended variant for additional dependency relations on top of

---

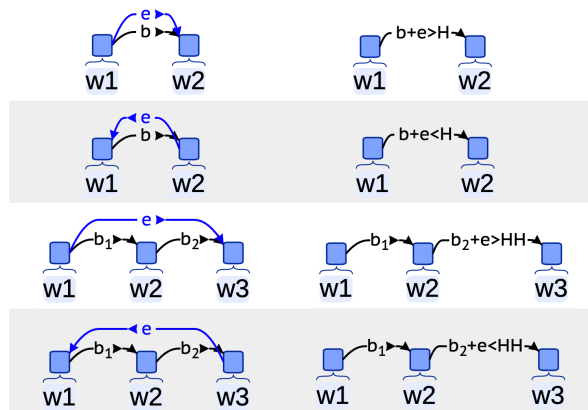
<sup>11</sup><https://huggingface.co/TurkuNLP/bert-base-finnish-cased-v1>

the base tree, and the Universal Dependencies scheme includes the enhanced representation for similar phenomena annotated in UD treebanks. Such relations help in having a shorter dependency path, in many cases a single relation, between interesting entities or other content-bearing words, and help to support certain downstream applications that benefit from representations capturing more aspects of the semantics.

Parsing into such graph representations has a long history with several approaches taken. Sagae and Tsujii (2008) introduced a simple extension of the transition system used in traditional transition-based dependency parsers (Nivre, 2004) supporting parsing into directed acyclic graphs (DAGs) rather than trees. While in the traditional transition system, the treeness is enforced by directly reducing the token from the system when an arc (relation) is created, the extended version defined additional transitions for creating an arc without reducing the token from the system, and thus allowing a second relation to be created later in the process. On the other hand, McDonald and Pereira (2006) approach the DAG parsing from the perspective of graph-based dependency parsers (McDonald et al., 2005), where the treeness is traditionally enforced by searching for a maximum spanning tree (MST) in directed, weighted graphs. In graph parsing work the strict MST decoding of graph-based parsing was replaced with approximate inference thus allowing output where a word may depend on multiple heads. In addition to dependency relation, parsing into graphs is also studied in the area of semantic parsing involving different semantic structures including e.g. proposition banks (Palmer et al., 2005) or abstract meaning representations (Banarescu et al., 2013), with several different parsing strategies taken.

While parsers capable of producing output not limited to the tree structure exist, most of the state-of-the-parsers implement the treeness requirement. In order to keep our pipeline as flexible as possible while producing enhanced graphs, and to be able to utilize any state-of-the-art, openly available parser also in the future, instead of parsing directly into the enhanced graph output, we considered two different approaches both implemented in terms of base trees: 1) encoding the enhanced representation into the base trees by enriching the set of dependency types, each dependency type encoding the base tree relation as well as additional enhanced relations when applicable, or 2) alternatively introducing the enhanced relations separately after the base parsing, creating a two-step approach. After preliminary experiments in Paper IV, we continued with the former approach. The overall approach of encoding the graph into the base tree is well-known and has been applied previously, e.g. by a number of teams in the SemEval tasks on semantic dependency parsing (Oepen et al., 2014, 2015).

In order to encode enhanced dependencies into the base tree, we concentrated on four different structures, which empirically measured cover the vast majority of the enhanced relations in existing UD annotations. The four encoded structures are



**Figure 5.** The four enhanced dependency structures captured in the encoding. The base (b) and enhanced (e) relations in the left column are encoded in a tree structure as in the right column. In the encoding, the symbol  $>$  stands for "relation from",  $<$  stands for "relation to",  $H$  is the head in the base tree, and  $HH$  is the head of the head in the base tree. [This figure was originally published in Paper IV.]

illustrated in Figure 5, from top to bottom being (1) an enhanced relation from base head to base dependent, (2) enhanced relation from base dependent to base head, (3) an enhanced relation from base head of head to base dependent, and (4) an enhanced relation from base dependent to base head of head. During encoding, the tree structure of the base tree is not changed, and all information is instead transferred to the base label names, indicating that in addition to the base relation, an enhanced relation should be created from  $X$  to  $Y$  with relation type of  $Z$ . This is also the case in situations where the direction or parent of the enhanced relation differs from the base relation as shown in the figure, where the direction is simply encoded with an arrow ( $<$  or  $>$ ) and two supported parent tokens with  $H$  (head in base tree) or  $HH$  (head of head in base tree). Even though this simple encoding scheme does not support arbitrary relations, in practise it's able to obtain near lossless representation for many UD treebanks, as will be shown later.

The downside of this approach is that the number of unique relation types increases substantially. While the above mentioned transformation gives us a label set of (base relation types \* enhanced relation types \* 4 enhanced structures), in practise the set of enhanced relation types is even more varied due to many lexicalised relation types, where the lemma of the head token or dependent token is encoded in the relation type (e.g. `conj:and` relation type in enhanced UD referring to the usage of coordinating conjunction *and* in coordination). In order to reduce the label set size in a language-agnostic fashion, we replaced the actual lemma with a placeholder encoding from which position (head or dependent) the lemma can be obtained during detransformation (e.g. `conj:d-lemma`).

The UD enhanced relations involve also representing empty nodes occurring in

elliptic constructions, where we again rely on the relation types and encode the empty nodes by following the procedure used in the shared task evaluations as well. In this encoding, the original relations involving an empty node  $e$  in between,  $(h, e, r1)$  and  $(e, d, r2)$ , are encoded with one direct relation  $(h, d, r1>r2)$  with both relation types. After all relations involving the empty node are encoded, the node can be removed. Similarly to others, this encoding is straightforward to reverse, the only exception is that the position of the original empty node is not stored, however, the position of the node is not determined to be relevant in the shared task and is not evaluated.

The final encoding is executed in the following order: 1) empty nodes 2) lexicalised relations are replaced with placeholders, 3) encode the four main relations showed in Figure 5, and 4) discard all remaining enhanced relations. After this procedure, we obtain a dependency tree, and the parser can be trained in a normal manner.

The encode-decode procedure can be evaluated with one transform–detransform cycle without an involvement of a parser, by first encoding the enhanced training graphs into trees, and then directly decoding them back, and measuring the Labeled Attachment Score on Enhanced dependencies (ELAS, F1-score over the set of enhanced dependencies in the system output and the gold standard) of the decoded data against the original. A lossless representation would result in ELAS of 100%. Across different UD treebanks, this value is in the 97.9–99.9% range, showing the encoding not being far from lossless, and only a minimal gain can be expected from encoding more complex structures. However, we note that this reflects the comparative structural simplicity of the enhanced relations present in the UD data, rather than the generality of our encoding. As a comparison, when measuring the ELAS using only gold base dependency trees as enhanced predictions, i.e. not trying to represent the enhanced relations at all and therefore measuring the oracle ELAS performance of a system limited to predicting only base trees, the score ranges between 67.3–98.0% across different treebanks (average being 80.7%), showing a clear drop compared to our encoding method.

During the prediction phase, a de-transformation of the encoding is executed in reversed order to obtain the desired dependency graph. However, there is a possibility of the parser producing a combination of encoded relations not leading to a valid enhanced representation after the de-transformation. Therefore, during the de-transformation all erroneous predictions producing invalid UD representation are deleted, possibly replacing the deleted relations with a valid approximation of the predicted relation in order to create a valid graph structure.

### 2.5.3 Turku Enhanced Parser Pipeline at the IWPT-2020 Shared Task

The IWPT 2020 shared task on Parsing into Enhanced Universal Dependencies (Bouma et al., 2020b) focused on evaluating several phenomena defined in the UD enhanced

representation. Such phenomena include gapping, coordination, control and raising constructions, relative clauses as well as case information, producing output expressed as dependency graphs, where a token may have multiple parent tokens rather than dependency trees, where only single parent is allowed for each token.

Similar to the CoNLL-18 shared task, the IWPT-2020 shared task evaluated the enhanced dependencies on top of predicted segmentation and morphological features, striving for systems starting from raw text and producing full dependency graphs. However, several intermediate steps, part-of-speech and morphological tagging, lemmatization, and base parsing into dependency trees, were optional in the shared task, and evaluated only as secondary metrics, the key focus being on enhanced relations.

**System Overview** For tokenization, multiword token expansion and sentence splitting we apply the Stanza toolkit by Qi et al. (2020), where all three are implemented as joint tagging over character sequences, where the model predicts end-of-token, end-of-sentence and end-of-MWT markers. After the identification step, MWTs are expanded as a separate step using a combination of frequency lexicon and neural sequence-to-sequence model. The models are obtained from the official released trained with each UD v2.5 treebank separately.

For part-of-speech and morphological tagging as well as dependency parsing we apply the multi-task Udify model initialized using weights from pre-trained language models. Udify implements separate prediction layers for each of the supported tasks learned jointly on top of the pre-trained BERT encoder. Before training the Udify models, the enhanced dependencies were encoded as relation types in the base tree using the encoding method described in the previous section. The Udify model is trained separately for each language instead of the original multilingual training.

For lemmatization, we use the Universal Lemmatizer described in Section 2.3 and Paper III, trained straightforwardly using the treebank data only.

As the shared task evaluation is carried out in terms of languages instead of individual treebanks, where all available treebanks for one language are merged together in the evaluation, for each language, we trained a model using the largest treebank (in terms of token count) in the shared task data release. All segmentation, tagging, parsing, and lemmatization models are thus monolingual and fine-tuned using data from a single treebank only. For each language the fine-tuning is based on a custom pre-trained BERT model selected based on the experiments reported in Section 2.5.1 and Paper IV.

**Results** The primary evaluation metric used in the shared task, ELAS, calculates F-score over the set of enhanced dependencies in the system output and gold stan-

dard.<sup>12</sup> Over the 10 teams submitting their systems in the time of the shared task submission, the Turku Enhanced Parser Pipeline achieved the highest performance in terms of the primary ELAS metric on average. For UD Finnish-TDT, our enhanced parser obtained 92% LAS and 89% ELAS, indicating high quality prediction of both base and enhanced relations. The accuracy of part-of-speech and lemmatization was 98% (UPOS) and 96% (lemmas). In terms of LAS, UPOS and lemmas, the performance increase is approximately +6pp, +2pp and +1pp respectively when compared with our CoNLL 2018 results, the biggest achievements coming from utilizing the pre-trained FinBERT language model.

In our preliminary experiments we measured whether the more complex labeling scheme used after the graph-to-tree transformation harms the prediction accuracy of the base dependency trees, and therefore the joint prediction of base and enhanced relation not giving an optimal accuracy for base trees. Most notably, when comparing the LAS of two near identical parsers, where one predicts only the base trees trained without including the enhanced dependencies in training, while the second is trained to predict the base trees where the labels include also information about the enhanced dependencies (the official IWPT submission), the accuracy of the base tree prediction (LAS) was nearly identical. Interestingly, the more complex relation labels therefore did not seem to harm the accuracy of the base relations.

## 2.6 Parser Utilization and Discussion

In this chapter we have presented work from four research papers, all building towards high-quality syntactic parsing pipeline of Finnish following the Universal Dependencies annotation scheme. The Paper I builds the essential groundwork for this research direction by introducing the first Finnish treebank available in the UD data collection, enabling the later participation in several highly popular shared tasks as well as many multilingual studies carried out on the UD data collection. The Papers II-IV concentrated more on technical aspects by studying different machine learning methods for morphosyntactic parsing. During these studies, we were able to substantially improve the accuracy of morphosyntactic parsing when measured on Finnish data. While our motivation mainly arose from the Finnish language, our contributions are highly multilingual, showing state-of-the-art results for a number of languages.

While majority of the academic work in the field of NLP strives towards developing highly accurate systems, other aspects of the software become crucial, particularly when deploying research software beyond its core developers. This is especially true in the area of morphosyntactic parsing, where many potential end-users, such as

---

<sup>12</sup>Note that in UD representation, most of the base relations are repeated in the enhanced representation if not explicitly deleted or replaced with another relation. Thus in ELAS metric, most of the "base relations" are taken into account as well.

researchers in linguistics or digital humanities, might not possess the same computational resources or skills as NLP developers. This creates certain limitations for research software if being deployed for end-users outside the NLP field. Given the concerns mentioned above, we have made several modifications to the components of the Finnish parser since the version detailed in this chapter and the accompanying papers. While the most accurate segmentation was achieved with models from the Stanza framework, during the shared task work the prediction speed of these models was found to be suboptimal. Although the performance difference between UDPipe and Stanza segmenters was significant, it was not substantial, leading us to choose the faster UDPipe component for large-scale deployment of the parser. Regarding computational skills, one concern is how easy the software is to install and use. While the usage of the parser is often relatively simple due to the provided top-level commands, installing cutting-edge research software that relies on multiple deep learning libraries can be challenging. We encountered this issue on several occasions when an update to one of these libraries caused failures in different parts of the pipeline. To address this we have done some modifications to the pipeline implementation (e.g. switching from Udify to Diaparser<sup>13</sup> and implementing a custom tagger), aiming for keeping the pipeline more stable and up-to-date. These kind of changes naturally have a small impact on the evaluation results, however, they do not substantially impact the outcomes as presented in this thesis.

Our publicly available Finnish parsing models has been used in several academic studies. To name few, for example Ivaska and Bernardini (2020) used the automatic analyses of the parser pipeline to study linguistic phenomena distinguishing constrained (non-native or translated) Finnish from non-constrained one, as well as distinguishing linguistic phenomena between different text registers. Kuusinen (2021) used the lemmas produced by the parser in the study of vocabulary diversity of Finnish learners. Janicki et al. (2020) on the other hand combined the parser pipeline into the workflow of news media analysis in the area of digital humanities. While the Finnish lemmatizer appears to be the most used function of the Finnish parsing pipeline, also other layers of analysis are utilized. For example, Ibrahim et al. (2019) utilizes the syntactic trees to identify common verbs and adjectives associated to different named entities in health related discussions, while Seuri et al. (2021) uses syntactic rules based on lemmas, part-of-speech tags, and syntactic relations for automatic extraction of direct and indirect quotations from news articles in media and political communication analysis. In addition to Finnish academic community, our parser has been used outside the academic circles as well in different public organizations or in the Finnish industry. However, measuring the software usage outside the academia is not straightforward due to these rarely leading to scientific publications.

---

<sup>13</sup><https://github.com/Unipisa/diaparser>

Several novel research directions in syntactic parsing have emerged subsequent to our work. For instance, Nguyen et al. (2021) introduced Trankit, a multitask learning framework based on adapter layers (Pfeiffer et al., 2020a,b), where the same underlying pre-trained language model, multilingual XLM-R model (Conneau et al., 2020) in their case, is fine-tuned for different tasks and languages by training several lightweight adapter layers while keeping the language model weights frozen. While the state-of-the-art results are continuously improving, their work is at least near the current state-of-the-art, especially when considering average performance over all UD treebanks. For Finnish, Trankit and the Turku neural parser pipeline give quite comparable results when taking into account the size of the pre-trained language model used.



## 3 From Structure to Meaning

While the structure of the language is a crucial element in successful communication, it does not uniquely convey the whole meaning. Altering sentence structure often influences meaning, and conversely, identical meanings can be expressed through a variety of grammatical constructs and lexical selections, i.e. paraphrasing. While many paraphrase pairs share some lexical cues, like overlapping words or phrases, they often undergo significant structural changes, as well as lexical synonym substitutions or the use of culture-bound idioms. This leads to cases where the paraphrase pair may not share any single surface unit, but is nevertheless completely (or at least almost completely) equivalent in meaning.

With the recent advancements in powerful, pre-trained language models for various traditional NLP tasks, there has been a growing interest in tasks that are more semantically oriented. Particularly, the recent progress has given hope of these models genuinely comprehending language rather than relying on simple surface cues. However, in several language understanding tasks the performance of these models have still been far from human capabilities, in many cases the model learning data artefacts rather than truly focusing on real semantics (Glockner et al., 2018; Tsuchiya, 2018; McCoy et al., 2019). In practical terms, we would like to have a well-established method capable of consistently generating highly similar representations for statements with equivalent contextual meanings but varying wording. The scarcity of deeply semantically-oriented resources in Finnish has slowed down such research direction. Thus we center our second set of research questions (RQ4 and RQ5) around this theme, and set out to study methods for constructing a high-quality corpus of Finnish paraphrases.

This chapter summarizes the creation of the Turku Paraphrase Corpus originally presented in Paper V. While the utilization of sentence-level heuristics is a prevalent approach for constructing large-scale paraphrase corpora, we hypothesize that these techniques might introduce a bias toward shorter and more straightforward instances that can be automatically recognized. To build a dataset emphasizing diverse paraphrases while avoiding selection bias towards short and lexically overlapping pairs, we study a novel manual paraphrase extraction approach. Employing this approach, we construct a corpus of over 100,000 paraphrase pairs, and proceed to compare its lexical diversity and length distribution with two paraphrase resources collected using automatic candidate identification. Additionally, we present baseline modelling

experiments conducted on the newly annotated data.

## 3.1 Building the Turku Paraphrase Corpus

A common way to construct a paraphrase corpus is to include a large amount of automatically gathered and labeled examples (with an optional manually curated evaluation section) (Dong et al., 2021; Wieting and Gimpel, 2018), or alternatively building a relatively small corpus of manually annotated examples (Dolan and Brockett, 2005; Lan et al., 2017). However, when targeting to fine-tune large language models on a paraphrasing task, many of the existing corpora are too small, while we hypothesize that the larger, automatically gathered datasets may introduce unwanted bias towards simpler and shorter paraphrases with higher lexical overlap. As many of the smaller, fully manually curated corpora include examples that are automatically collected using different heuristics (Dolan and Brockett, 2005), the manual curation may not be enough to remove the bias possibly introduced during candidate selection.

In the collection of the Turku Paraphrase Corpus we set out to gather a large-scale, over 100,000 example corpus of high-quality Finnish paraphrase pairs with relatively low surface similarity. In order to avoid possible candidate selection bias, we rely on manual candidate selection, where an annotator receives two text documents, and extracts all interesting paraphrase candidates from the document pair, later also labeling each pair based on the developed annotation scheme. Next, the paraphrase data collection and annotation is briefly discussed, and then we will continue to evaluate our corpus from several aspects and compare its average paraphrase length and lexical similarity with two other paraphrase corpora available for Finnish. Finally, we briefly show several experiments where the paraphrase data is used in different applications.

### 3.1.1 Manual Paraphrase Extraction

In the manual paraphrase extraction phase, an annotator receives two text documents, and extracts all interesting paraphrase candidates from the document pair. The documents are selected to be paraphrase-rich, i.e. have a high probability for naturally occurring paraphrases in order to secure a sufficient yield of paraphrases given the time invested to the annotation. However, we aim to sample as many different text sources as possible, at the same time optimizing the usage of person-months available for annotation, especially keeping in mind the aim of gathering a corpus including over 100,000 high quality paraphrase pairs. We experiment with five different text sources: 1) alternative Finnish subtitles for the same movies or TV episodes, 2) news headings and articles discussing the same event in two different Finnish news sites, 3) different messages with identical title and sub-forum information from a popular Finnish discussion forum, 4) alternative student translations from university transla-

tion courses, and 5) alternative student essays from university course exams. Next, we shortly describe each text source, and then continue to the extraction workflow and statistics.

**Alternative subtitles** OpenSubtitles<sup>1</sup> provides a large amount of user-generated subtitles for various movies and TV episodes. The subtitles are available in several languages, and oftentimes there are several independently translated versions for a single language. This offers an opportunity to make use of the natural variation in language produced by independent translators by pairing two Finnish subtitle version for a single movie or TV episode. In total, subtitles for over 2,700 movies and TV episodes were used for building the corpus, however, only a randomly selected, relatively short segment from each subtitle pair was annotated in order to avoid a large number of unnecessarily close examples.

**News articles and headings** News articles published by two different Finnish news agencies approximately at the same time have a high likelihood of reporting on same events, and therefore including similar statements. While sometimes both news agencies are releasing a near-identical article obtained from a third party source, some of the news articles are written independently by the two agencies including legitimate paraphrasing. We utilize news articles collected from both YLE (the national broadcaster) and Helsingin Sanomat (Helsinki News) RSS feeds during the years 2017–2020. A total of 2,700 full-text articles and 1,500 news headings pairs were used in the paraphrase extraction.

**Discussion forum messages** We hypothesize that different discussion forum messages related to same topics include a number of naturally occurring paraphrases. For example, different thread-starting messages under the same subforum often seek information about the same topic or share related experiences. Similarly, different replies to the same message often convey similar reactions. We set out to experiment with thread opening messages from the Suomi24 discussion forum<sup>2</sup> posted into the same subforum and having an identical title. We were able to find both highly similar messages written by different people, but also paraphrased messages clearly originating from the same author. A total of 7,300 message pairs proceeded into the paraphrase extraction, however, many of the document pairs did not produce any paraphrases in the end.

---

<sup>1</sup><http://www.opensubtitles.org>

<sup>2</sup><http://urn.fi/urn:nbn:fi:lb-2019021101>

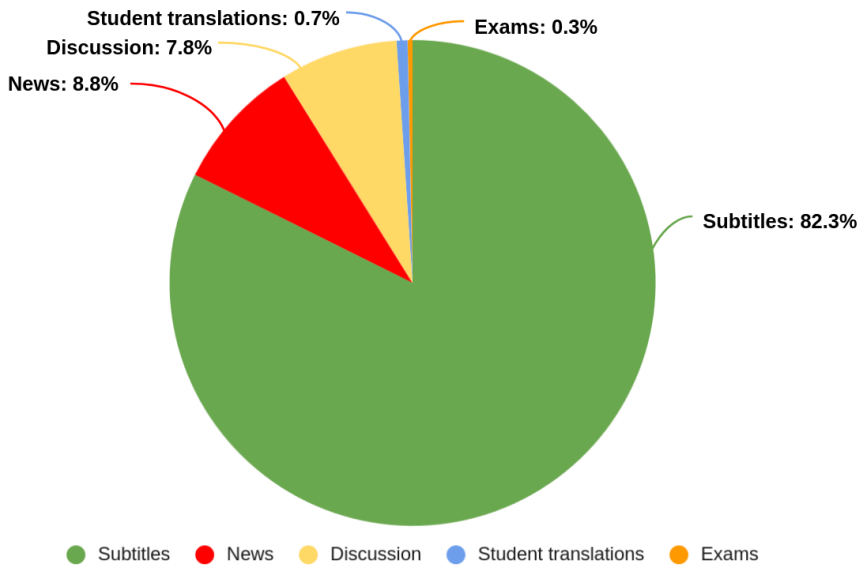
**Student translations** In addition to alternative subtitles, we also initiated an attempt to find other source materials where the same foreign text is translated into Finnish by multiple translators. One potential source of constant amount of alternative translations is exercises from different language studies and courses, where several people translate the same exercise text. We targeted exercises taken from the university courses in translation studies. Unfortunately, due to the requirement of a written consent of each student, which is difficult to secure, we were able to gather only 16 unique exercise texts including at least two different student translations with the appropriate permission.

**University exams** The final text source experimented with is student essays collected from the university course exams, where the hypothesis is that essays answering the same exam assignment will include similar arguments, and therefore have a high probability for naturally occurring paraphrases. Due to the similar restriction than in student translations, we were able to collect a total of 24 unique exam questions or essay assignments for which at least one candidate pair (two alternative essay answers) was available.

Given a candidate document pair, the overall manual annotation workflow starts from paraphrase extraction, where an annotator sees the two text documents side-by-side in a custom tool designed for fast paraphrase extraction. The annotator can independently scroll the two documents, and extracts a paraphrase pair by selecting the corresponding text passages from both documents. Annotators are instructed to select all interesting paraphrase candidates, where a paraphrase can be anything between a short phrase and a long text segment including multiple sentences, therefore the paraphrase extraction is by no means limited to follow sentence boundaries. Candidates including only uninteresting variation, such as minor differences in inflection and word order, are avoided during extraction in order to collect a corpus with non-trivial examples.

One advantage of manually extracting paraphrases from their original documents is not only ensuring the quality of the examples but also the opportunity to gather and assess the paraphrase candidates within their native context. During the extraction, in addition to storing the paraphrase pair, we also save the information of where in the document the text appeared, giving us the unique opportunity to study paraphrasing in context rather than merely comparing two standalone segments as is the case in most if not all paraphrase corpora. To our knowledge, this work is the first large-scale paraphrase corpus providing original document context information for the paraphrase pairs, setting many new possibilities for contextual paraphrase recognition.

In Paper V we compare paraphrase extraction rates of different text sources from many different aspects. In summary, the two translation based sources (alternative



**Figure 6.** The amount of paraphrase pairs in the Turku Paraphrase Corpus originating from different text sources.

subtitles and student translations) end up being the most time-efficient source materials. While the alternative translations tend to include the statements in similar order, the other materials may include similar statements but in any arbitrary order, making it more difficult to find the corresponding statements, and thus, slowing down the paraphrase extraction. Out of these two translation based sources, the student translations were clearly the most productive text source experimented with in terms of time-efficiency, the high average number of paraphrase candidates extracted from a document pair, and the low number of documents pairs not producing any candidates (more details given in Paper V). However, due to the limited amount of student materials available, we had to mainly concentrate on the three other text sources. From these three sources (subtitles, news, and discussion forum messages), subtitles were by far the most efficient in terms of annotation time, and thus this source is over-represented in the corpus. The proportional amount of paraphrase pairs obtained from different text sources is summarized in Figure 6, the alternative subtitles dominating the final dataset with 82% share, news texts and discussion forum messages both having bit less than 10% portion both, while both student materials represent only a tiny fraction of the corpus. To conclude the manual extraction utilizing different text sources, the student produced materials were found promising in our experiments, however, the work required to settle legal restrictions on these prevented their large-scale utilization and therefore student based materials do not greatly contribute to the final corpus.

### 3.1.2 Paraphrase Annotation

After the candidate extraction, all candidate paraphrases are manually annotated according to the annotation scheme developed as part of the corpus construction. The annotation scheme consists of the base scale 1–4, where labels 1 and 2 are used for negative candidates (1: unrelated, and 2: related but not a paraphrase), while labels 3 and above are paraphrases at least in the given context if not everywhere (3: context dependent paraphrase, and 4: universal paraphrase) with additional subcategories (flags) for distinguishing different types of paraphrases which would otherwise fall from the label 4 into label 3.

While similar numeric scheme to our base scale is used in many earlier studies to express the scale between unrelated sentences and full paraphrases (see e.g. Dong et al. (2021) and Creutz (2018)), explicitly defined finer subcategorization is not widely adopted in paraphrase annotation. Bhagat and Hovy (2013) defines several categories of near paraphrases, which in practice are considered as paraphrases in many NLP studies, however their scope of study is not to define a practical annotation scheme. Nevertheless, some of the categories mentioned in the study comply with our subcategorization. Our full labeling scheme is shown in Table 5 together with example annotations.

The novel notion in the annotation scheme is its ability to separate universal paraphrases in all context (*label 4*) and context dependent paraphrases being paraphrases in the given context, however, not necessarily everywhere (*label 3*). The main reasons for context dependent paraphrases are sentence ambiguities as well as different specificity or minor details, where for example a pronoun or other reference is clear from the context, however, the same reference does not hold in all contexts. Similar to (Gold et al., 2019), who studied interaction between different semantic relations such as paraphrasing and inference, we also noticed the occasional unidirectionality in paraphrasing, where the paraphrase pair could be considered universal in one direction, however, not to both directions. In order to account this behavior, we introduced the subsumption flag between the universal and context-dependent paraphrases to mark cases where the substitution test of universal paraphrases holds for one direction, but not to the other, i.e. the more detailed statement can always be replaced with the more general one without losing the principal meaning, but the more detailed one does not fit all contexts where the general statement could be used. Common cases of unidirectionality are pairs where one of the two statements is ambiguous while the other is not, or one including a minor additional detail the other is omitting. If there is a justification for crossing directionality (one statement being more detailed in one aspect while the other in another aspect), the pair falls into context-dependent paraphrase (*label 3*) as the directional replacement test does not hold anymore.

In addition to subsumption, there are also two other flags used in the corpus. The

Label	Definition	Example
4	<b>Full (perfect) paraphrase</b> in all reasonably imaginable contexts, one can always be replaced with the other.	Tulen puolessa tunnissa. Saavun 30 minuutin kuluessa.  <i>I'll be there in half an hour.</i> <i>I will arrive in 30 minutes.</i>
4 </>	<b>Subsumption: one of the statements is more detailed and the other more general.</b> The relation is directional, the more detailed statement can be replaced with the more general one in all contexts, but not the other way around. Arrow 'points to' the more general one.	Haluan vain puhua opettajalle. Tahdon vain puhua hänen kanssaan.  <i>I just want to talk to the teacher.</i> <i>I just want to talk to him/her.</i>
4 s	<b>Tone or register: the meaning of the two statements is the same, but the statements differ in tone or register</b> such that in certain situations, they would not be interchangeable.	Päivää työt! Helou gimmat!  <i>Good day, girls!</i> <i>Hey, you gals!</i>
4 i	<b>Minor deviation: minimal differences in meaning or easily traceable differences in grammatical number, person, tense or such.</b> The treatment is considered application dependent.	Tämä laite on epäkunnossa. Tuo kone on rikki.  <i>This apparatus is malfunctioning.</i> <i>That machine is broken.</i>
3	<b>Context dependent paraphrase</b> , the meaning of the two statements is the same in the present context, but not necessarily in other contexts.	911. Hätänumero.  911. <i>Emergency number.</i>
2	<b>Related but not a paraphrase</b> , a clear relation between the two statements, yet they cannot be considered paraphrases.	Kadonnut 12-vuotias poika löytynyt. Poliisi etsii 12-vuotiasta poikaa.  <i>The missing 12-y boy has been found.</i> <i>The police are searching for a 12-y boy.</i>
1	<b>Unrelated, there being no reasonable relation between the two statements</b> , most likely a false positive in candidate selection.	Oletteko Sherlock Holmes? Riippuu.  <i>Are you Sherlock Holmes?</i> <i>It depends.</i>
x	<b>Skip:</b> labeling a candidate pair is not possible for a reason or giving a label would not serve the desired purpose (e.g. wrong language or identical statements).	Mikä nimesi on? Vad heter du?  <i>What is your name?</i> <i>Vad heter du?</i>

**Table 5.** The annotation scheme used in the Turku Paraphrase Corpus. Examples annotated with label 1 (unrelated) or label x (skip) are discarded from the final corpus.

Original	
Voinko palata tehtäviini?	Can I get back to my assignments?
Saanko jatkaa?	Can I continue? 4>
Rewrite	
Voinko palata tehtäviini?	Can I get back to my assignments?
Saanko jatkaa <i>tehtäviäni</i> ?	Can I continue <i>working on my assignments</i> ? 4

**Figure 7.** An illustration of one rewritten paraphrase pair taken from the corpus. Modifications are shown using emphasized font.

style flag (s) marks for tone or register differences in cases where the meaning is in principle the same, but the way of saying is different in a manner such that the statements would not be fully interchangeable e.g. between different text registers. The minor deviation flag (i) on the other hand tracks for certain systematic differences often occurring in the data, which create minor deviation to the meaning (such as *this* vs. *that*), however, which in many use cases or downstream applications could easily be ignored. However, in respect to other applications, such as paraphrase generation, it could be desirable for the system to not learn to introduce such differences.

All flags are independent of each other (disregarding the directional subsumption flag) and can be combined in the annotation, however, the flags are only attached to label 4 paraphrases (subtypes of universal paraphrases). If the paraphrase pair is already annotated as context dependent (label 3), the styling and other flag differences are not considered anymore, as the contextuality itself already covers such differences.

After the manual extraction, all paraphrase candidates are transferred into a custom label annotation tool, where the annotator can assign a label for each candidate pair separately. In this tool, the paraphrase candidates are shown individually, but including the original document context in order to support contextual annotation. In addition to paraphrase labeling, the tool provides an option for rewriting each paraphrase pair to be fully interchangeable, universal paraphrases, if the original labeling indicates the candidate not to be such. The annotators are instructed to rewrite non-universal paraphrase pairs in cases where a simple edit, for example word or phrase deletion, addition or re-placement with a synonym or changing an inflection, can be easily constructed. Rewrites must be such that the annotated label for the rewritten example is always label 4. In cases where the rewrite would require more complicated changes or would take too much time, the annotators are instructed to move on to the next candidate pair rather than overthink the possible rewrite options. An example of one rewritten paraphrase pair from the corpus is demonstrated in Figure 7.



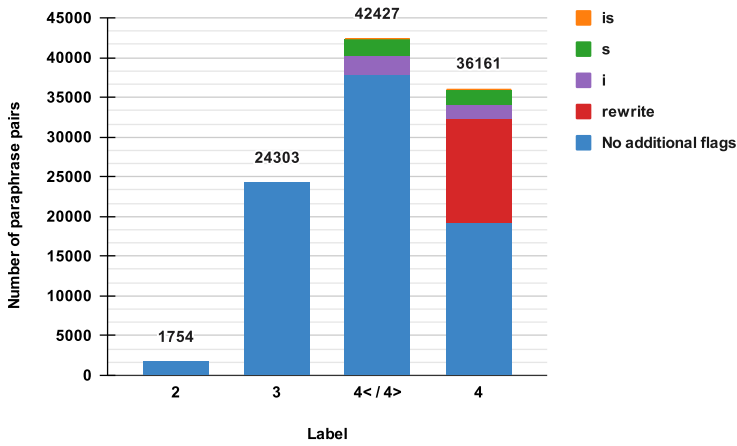
Section	Examples	Rewrites	Total
Train	73,165	10,480	83,645
Devel	9,231	1,298	10,529
Test	9,208	1,263	10,471
Total	91,604	13,041	104,645

**Table 6.** Data sizes in the Turku Paraphrase corpus. [This table was originally published in Paper V.]

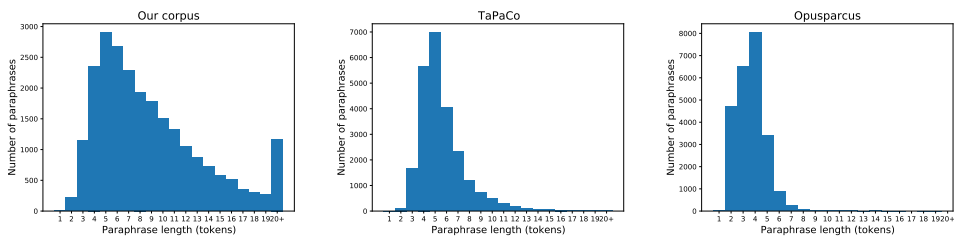
## 3.2 Corpus Statistics and Evaluation

**Statistics** The total number of paraphrase pairs in the final corpus is 104,645, divided into training, development and test sections using roughly 80/10/10 split. In the Table 6 we illustrate separately for each section the number of paraphrases directly extracted from the source documents, as well as the number of human-made rewrites, the rewrites representing 12% of all paraphrases in the final corpus. The label distribution of the corpus is illustrated in Figure 8, showing a mere 2% of all annotated paraphrases being labeled as negative, i.e. not representing a paraphrase pair. The negative pairs consist of examples which are related but not paraphrases (label 2 in the annotation scheme), as the small number of pairs determined to be unrelated (label 1 in the scheme) were discarded from the final corpus. While the actual paraphrase distribution is difficult to compare across different paraphrase corpora due to many of them lacking manual annotation, we carry out a label distribution comparison on those including full manual annotation, Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005), PARADE (He et al., 2020), ParaPhraser (Pivovarov et al., 2018), and Quora Question Pairs<sup>3</sup> (QQP). The outcome of the analysis shows our corpus standing out from the other resources in respect of the label distribution, while our corpus includes 98% of positive examples, the ratio is 67% in MRPC (binary annotation), 47% in PARADE (binary annotation derived from the original 4 labels scheme in the official release), 23% in ParaPhraser (annotation with three labels, 64% if taking into account also those labeled as having similar meaning instead of taking into account only pairs annotated as having same meaning), and 37% in QQP (binary annotation). While the heavily skewed distribution towards positive paraphrases include the obvious advantage of having more true paraphrase pairs for various experiments, it also poses new challenges. For example, when training a classifier for binary paraphrase detection with labels paraphrase or not-a-paraphrase, it’s expected that a sufficient amount of negative examples are available. However, our hypothesis is that it’s better to invest the expensive human resources into finding high quality positive examples, rather than manually annotat-

<sup>3</sup><https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs>



**Figure 8.** Label distribution in our corpus. Here *is*, *s*, *i* refer to the various additional flag combinations. [This figure was originally published in Paper V.]



**Figure 9.** Comparison of paraphrase length distributions in terms of tokens per paraphrase. [This figure was originally published in Paper V.]

ing a large amount of negative examples.

**Annotation agreement** To ensure the consistency of the label annotation and measure annotation agreement, approx. 2% of the paraphrase pairs in the corpus are double annotated. In double annotation two annotators annotate the labels independently from each other for the same paraphrase candidates, and the annotations are merged and conflicting labels resolved together with the annotation team, resulting in a consolidated subset of consensus annotation. When measuring the individual annotations against the consensus labels, the overall accuracy is around 70%, when using the full set of labels used in the annotation. The level of agreement is on par with similar numbers reported in other paraphrase studies (Dolan and Brockett, 2005; Creutz, 2018).

Word unigrams		Character trigrams	
sim 0.0	16%	sim 0.0	3%
sim 0.2	38%	sim 0.2	16%
sim 0.4	68%	sim 0.4	44%
sim 0.6	90%	sim 0.6	79%
sim 0.8	99%	sim 0.8	98%

**Table 7.** Cumulative percentage of paraphrase pairs falling below different similarity cutoffs measured on the corpus training section.

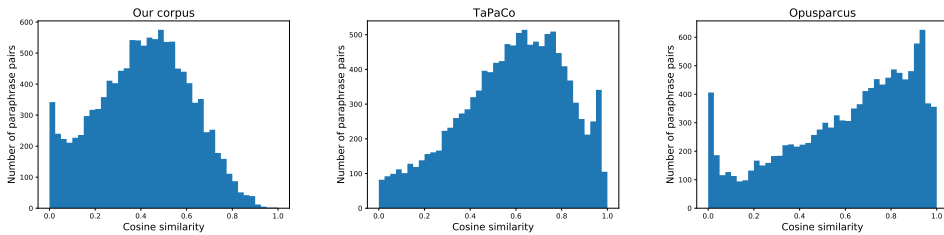
**Paraphrase length** In order to test our hypothesis of manual paraphrase extraction producing longer and more varied examples compared to automatically identified candidates, we compare our corpus with two other paraphrase resources available for Finnish. Opusparcus (Creutz, 2018) provides 3,700 manually labeled paraphrase pairs for Finnish with an additional release of automatically scored and filtered candidates with different quality threshold ranging from 480K to few million candidates. TaPaCo (Scherrer, 2020) includes 12K paraphrase candidates for Finnish without any manual verification. Both are multilingual datasets where Finnish is one of the included languages. To keep the sizes of the compared sets uniform, we sample 12,000 examples from each corpus for the study. Figure 9 illustrates that the paraphrase length distribution in our corpus is broader, including not only shorter paraphrases but also a noteworthy proportion of longer ones. In contrast, the other two corpora predominantly consist of relatively concise paraphrase candidates. On average, our corpus contains 8.8 tokens per paraphrase statement, compared to 5.6 in TaPaCo and 3.6 in Opusparcus.

One positive outcome of the manual paraphrase extraction is that it does not restrict the selection to follow sentence boundaries or any other predefined units, but gives a possibility to select text segments of any length. Next, we measure what is the distribution of short phrases, single sentences, or longer than a sentence units. To this end, we apply our Finnish dependency parser pipeline to segment sentence boundaries and recognize whether a paraphrase is a well-formed sentence (starts with a capitalized letter, ends with a punctuation and includes a main verb) or not. The outcome is that approximately 12% of the paraphrase statements are phrases or units not resembling a well-formed sentence, 73% are well-formed, single sentences, 13% are two sentences long, and the remaining 2% being more than two sentences long segments. When looking into paraphrase pairs instead of individual paraphrase statements, 63% of the pairs have one-to-one mapping of well-formed sentences, following with one-to-two (10%), sentence-to-phrase (9%), phrase-to-phrase (7%), and two-to-two (7%) mappings, the other variants occurring only rarely.

Statement 1	Statement 2
<b>Word unigrams</b>	
Tämä on viimeinen pyyntöni. (This is my last request.)	En pyydä enää mitään muuta. (I don't ask for anything else anymore.)
Todennäköisesti tekolintu. (Probably a mechanical bird.)	Luultavasti joku keinotekoinen lintu. (Probably some kind of artificial bird.)
Ajatuksesi näyttää harhailevan. (Your thoughts seem to wander.)	Olet kuin muissa maailmoissa. (You are like in other worlds.)
<b>Character trigrams</b>	
Menkää rappusista. (Go by the stairs.)	Kulkekaa portaita pitkin. (Walk up the stairs.)
Se tietää pitkää kakkua. (That's a ticket to jail.)	Siitä saa paljon linnaa. (That's a long stretch.)
Upeaa. Tosi upeaa. (Fantastic. Really fantastic.)	Kymmenen pistettä ja papukaijamerkki. (Full points and a medal.)

**Table 8.** Example paraphrase pairs with annotated label 4 and zero lexical overlap in terms of word unigram or character trigram features. The examples are selected from the corpus training section.

**Lexical similarity** One of the main objectives of our corpus creation work was to avoid populating the corpus with a high amount of lexically very similar, and therefore likely uninteresting, paraphrases. In order to study how lexically dissimilar our manually extracted paraphrases are, we measure the lexical similarity in terms of tf-idf weighted cosine similarity using both word unigram and character trigram feature representations. While the word-level measure directly accounts for sharing the exactly same tokens, the character trigram features also tries to account for the different inflectional variants of the same base words. When using the word unigrams feature representation, the mean similarity is 0.3 with 16% of the paraphrase pairs not sharing even a single word (cosine similarity 0.0). When measuring using character trigrams, the mean cosine similarity is 0.4, with 3% of the pairs not having a single shared character trigram. In Table 7 we report the cumulative percentages using different similarity cutoffs, and in Table 8 paraphrase pairs with zero lexical similarity are demonstrated. In Figure 10, we plot the lexical similarity distributions for three paraphrase datasets available for Finnish. This comparison reveals that indeed our corpus contains a higher percentage of paraphrases exhibiting lower lexical similarity, whereas the distribution in the case of the other two corpora tends to favor pairs with greater lexical overlap.



**Figure 10.** Comparison of paraphrase pair cosine similarity distributions. [This figure was originally published in Paper V.]

### 3.3 Discussion and Paraphrase Modelling Experiments

Given the Turku Paraphrase Corpus, we now have the possibility to train and evaluate models on a task closely associated with Finnish semantic understanding. The ability to recognize and understand paraphrases is crucial for various downstream applications. For example, one of the tasks benefiting of such understanding is a project of university essay grading support, where the objective is to develop and evaluate methods for supporting grading of Finnish university essays. (Chang et al., 2021b) Aside from plain autograding, such as predicting a grade for the given essay, one possible option to support grading of textual essays is to highlight argumentations from student essays similar to reference materials (e.g. a reference answer, or already graded essay) in order to speed up the manual essay evaluation. In such cases the surface lexical realizations (words or sentence structure) may not be sufficient enough, if the students are addressing similar arguments using different wordings. Instead, a deeper paraphrase understanding is required.

In Paper V we present baseline results for two different paraphrase modelling experiments; paraphrase classification based on a pairwise label classifier, and paraphrase retrieval based on fine-tuned sentence embeddings. In addition to these, in Kanerva et al. (2024) we utilized the contextual information of the paraphrases and introduced a novel paraphrase task setting, where the objective is to extract a text span from the given source document constituting a paraphrase of the query statement. This span detection approach is highly inspired by similar work on semantic search and question answering. Next these experiments are briefly summarized to illustrate the possible ways to utilize the corpus.

#### 3.3.1 Paraphrase Classification

The manually annotated labels in the paraphrase corpus can be straightforwardly used to train a supervised paraphrase classifier. The classifier is implemented as pairwise multi-output model, where the model receives one candidate pair at a time, and predicts a label for it. The model output consists of four jointly trained prediction

Label	Prec	Rec	F	Support
2	40.2	32.9	36.2	161
3	59.3	52.6	55.8	2434
4<	56.0	58.1	57.0	2003
4>	58.3	59.8	59.1	2287
4	70.5	73.9	72.2	3586
i	51.8	48.9	50.3	454
s	49.4	37.7	42.8	438
W. avg	57.9	58.3	58.0	
Acc			58.3	

**Table 9.** The paraphrase classifier trained and evaluated on the Turku Paraphrase Corpus. [This table was originally published in Paper V.]

layers, one for the base label (with classes 2, 3 or 4), one for the subsumption flag (<, > or none), one for the style flag (s or none), and one for the minor deviation flag (i or none).

The classification results are shown in Table 9, where the results are tabulated separately for labels 2, 3, 4<, 4>, and 4 (disregarding the flags s and i) as well as for the flags s and i (disregarding the base label). Together with these, the weighted average F-score and accuracy of full labels are reported. The initial modelling results verifies the challenging nature of the dataset, giving weighted average F-score of 58%. The small amount of label 2 paraphrases in the corpus clearly poses a challenge for the classifier with an F-score of only 36%. In addition to their rarity, the label 2 examples are likely serving as extremely difficult negative examples due to the fact of these examples passing the strict candidate selection during annotation.

### 3.3.2 Sentence Embeddings

In addition to classification, paraphrasing is often estimated measuring semantic relatedness through embedding similarity, where the embeddings are created such that paraphrased statements receive a high similarity score (e.g. cosine similarity or euclidean distance), while semantically unrelated statements receive a low similarity score. The advantage of similarity based methods compared to pairwise classifiers is them being considerably cheaper in terms of computational time when applying to many real-life scenarios including millions of candidate sentences. While the classifier needs to embed each pair separately, the embeddings can be calculated once for each statement while the pairwise comparison can be obtained by running only a lightweight similarity function on top of the pre-calculated embeddings.

We use the test set of the Turku Paraphrase Corpus to evaluate three different

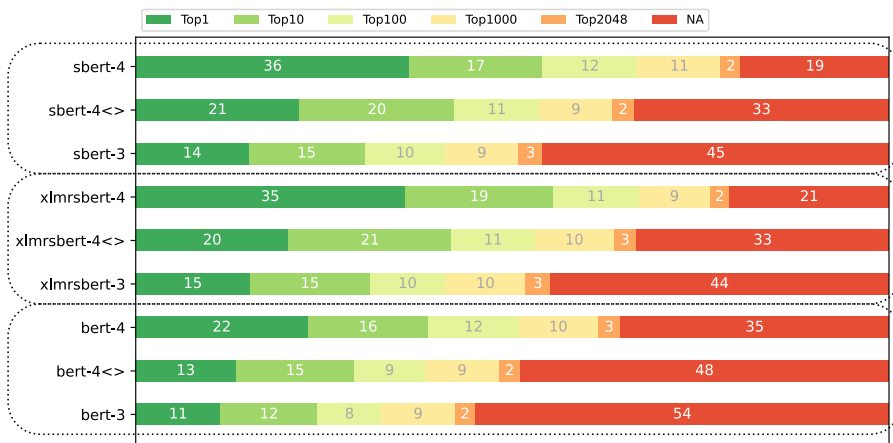
models on paraphrase retrieval using embedding similarity.

**The vanilla FinBERT** without any task specific fine-tuning. The embedding for the given statement is the average of the token embeddings obtained from the final BERT layer.

**The Finnish SBERT model**, where we fine-tune the FinBERT language model for paraphrasing using Sentence-BERT (SBERT) training objective (Reimers and Gurevych, 2019), where the aim is to improve individual sentence embeddings in order to better support the direct cosine similarity comparison. In addition to the Turku Paraphrase Corpus training set, a large amount of positive and negative paraphrase candidates are automatically collected for fine-tuning (more detailed description available in Paper V).

**The multilingual SBERT model** (paraphrase-xlm-r-multilingual-v1) by Reimers and Gurevych (2020), where they first fine-tune a monolingual English SBERT model using different datasets including semantically similar English sentence pairs. This model is then used as a teacher model in multilingual knowledge distillation (teacher–student framework) where the multilingual XLM-R language model (Conneau et al., 2020) is fine-tuned to mimic the embeddings of the teacher model on parallel data for over 50 languages. Therefore, the multilingual fine-tuning aligns the multilingual space to follow the semantics of the paraphrase fine-tuned English embedding space.

The evaluation results are shown in Figure 11, where we simulate a realistic paraphrase mining setting by testing the retrieval of the target among a set of 399M unique sentences acting as distractors for the correct target. In this evaluation setting, we first embed each sentence, and then, for each test set paraphrase pair  $(s_1, s_2)$ , we measure the similarity of  $s_1$  against all other sentences and calculate at which rank out of the nearly 400M candidates the embedding of  $s_2$  is found in terms of Euclidean distance of normalized embeddings. The results are reported for top N values of 1, 10, 100, 1000, and 2048, i.e. measuring how often the model ranks the correct paraphrase statement among top-N out of the 400M candidates. The Finnish SBERT model is able to retrieve 14–36% of the paraphrases as a top-1 candidate depending on the paraphrase label, while a total of 29–53% are returned among top-10 candidates, demonstrating to be a highly efficient method even in a case of retrieving from a massive set of candidate sentences. When comparing different models, the vanilla FinBERT without task specific fine-tuning performs clearly worse compared to the two SBERT models, while the Finnish SBERT and the multilingual SBERT perform roughly on-par, both being trained on paraphrase data (either Finnish or English), however the multilingual model being exposed to an additional massive set of translation pairs as well.



**Figure 11.** The retrieval of test set paraphrase pairs by different models. The retrieval is measured for the three main classes of paraphrase separately (4, 4< or 4>, and 3) disregarding flags *s* and *i*. The colors indicate different top *N* values, where *NA* means that the correct sentence did not rank in the top 2048 list, which was the upper technical limit in the experiment. The numbers on top of the bars indicate the number of returned paraphrases in percentages. [This figure was originally published in Paper V.]

### 3.3.3 Paraphrase Span Retrieval

In Kanerva et al. (2024), we approach the task of semantic search by defining paraphrase detection as extractive span detection from the given source document. This novel task setting utilizes the contextual information of the paraphrases available in the Turku Paraphrase Corpus, and studies how well a span detection model is able to extract the correct span constituting a paraphrase for the given search statement. The task setting is similar to question answering (QA), where for the given question, the model is trying to extract the span from the given source document constituting a relevant answer for the input question. In our setting, the question–answer pairs are replaced with paraphrase pairs, both being considered as a subtask under the semantic search framework. In QA, the objective is to directly extract the answer for the posed question. In our paraphrase span detection, the objective is then to find the relevant information from the source document even if expressed using different wording compared to the search phrase. Altogether, a complete semantic search system can be considered to be composed of three components: (1) candidate document retrieval system able to return all relevant documents for the given search phrase from a massive collection of source documents, (2) QA component extracting the actual answer from the candidate documents retrieved by the candidate retrieval system in cases where the search query is expressed as a question, and (3) paraphrase span detection component extracting the relevant passages from the candidate documents returned by the candidate retrieval component in cases where the relevant outcome is information expressing the same meaning despite of the actual words used.



In the scope of this paper the aim was not to build a complete system for semantic search, but rather to evaluate the paraphrase span retrieval component in isolation. By giving the correct source document, we evaluated how likely the span detection model is able to retrieve the correct span from the document. The analysis in the paper suggested the span retrieval model being superior to methods based on fine-tuned sentence embeddings, as well as the span detection model trained on the Turku Paraphrase Corpus being remarkably better compared to the same model trained on silver-standard paraphrase data created through automatic back-translation.

## 4 Conclusions and Future Work

This thesis has described the development of several NLP resources for the Finnish language, beginning with structural morphosyntactic analysis and progressing towards the paraphrasing task, more meaning-oriented in nature. Alongside the elementary data creation work, the described datasets were used to develop different machine-learning approaches aimed at analyzing and comprehending the structure and meaning of the Finnish language. In several instances, we achieved state-of-the-art results at the time of publication.

Four of the papers included in the thesis (Papers I, II, III and IV) collectively contribute to a highly accurate syntactic parser for Finnish, addressing the research questions RQ1, RQ2 and RQ3. When starting to work on this thesis, we already had a reasonably sized, manually annotated treebank for Finnish at our disposal, yet the parsing numbers were moderate compared with those published for some other languages. Therefore, in RQ1 we set out to study whether the performance was limited by the annotated data, the technology, or the language itself by asking *whether Finnish is inherently more challenging to parse with regards to accuracy when compared to other languages, such as English, and how far can we advance in dependency parsing without the necessity to increase the size of the manually annotated corpus*.

When publishing the very first results for Finnish UD parsing in Paper I, the labeled attachment score was around 82%. However, throughout the work presented in this thesis, as well as other contributions introduced during the thesis work by our research group or wider research community, to our knowledge the best labeled attachment score published for the UD Finnish-TDT corpus is around 92–94% depending on the size of the pre-trained language model used. Altogether, a very substantial improvement of approx. 10pp (absolute) is obtained during the period of the thesis publications. When contrasting the obtained numbers for human-performance on the task, the closest point of comparison can be obtained from Haverinen et al. (2014). The paper reports the labeled attachment score between an individual annotator and consolidated consensus annotations<sup>1</sup> being between approx. 88–96% depending on the annotator. The parsing numbers obtained with the latest pre-trained language models are thus narrowing the gap between human and model performance, if not

---

<sup>1</sup>Double annotated data, later merged and conflicts resolved together with the whole annotation team.

already reaching on average human performance on the task. When reflecting this to the first research question, an interesting outcome here is that these improvements were obtained without increasing the size of the manually annotated corpus used in training but rather relying on methodological contributions and language model pre-training. Therefore, to conclude regarding the RQ1, with the existing TDT dataset we are able to achieve at least near human performance on dependency parsing when evaluated on relatively clean and standard Finnish language. Additionally, when considering the overall performance of the Finnish parser, as well as comparing the performance of same methods applied to different languages, Finnish does not appear to be substantially more difficult to parse compared to others as its overall performance is on par or even above the generally expected level.

Methodologically, when reflecting on RQ2 asking *what methodological approaches should be employed to optimize the accuracy of the parsing pipeline*, there were two bigger advancements in the dependency parsing performance for Finnish; utilization of Dozat’s parser (Dozat and Manning, 2017) in Paper III and integration of the pre-trained FinBERT language model in Paper IV. The Dozat’s parser builds a powerful feature representation by utilizing pre-trained word embeddings contextualized using bidirectional LSTM layers. This method reached approx. 87% LAS and substantially outperformed both our earlier feature-based parser used in Paper I, as well as the first neural dependency parser trained for Finnish by Straka et al. (2016). A similar feature extraction architecture was shown to match or surpass state-of-the-art results also in other studies, e.g. Kiperwasser and Goldberg (2016). In contrast to the previous generation of feature-engineered parsers, this approach has several advantages in addition to performance improvement. Albeit being computationally complex, the feature representation is simplified in a way that there is not a need for hand-engineering features for different language phenomena anymore. This also means that languages do not need to be treated separately based on their properties, but to some extent the same model can be applied to several (if not all) languages that has sufficient amount of training data available.

The second big advancement towards human performance in dependency parsing can be attributed to the large-scale pre-training regime. While Dozat’s parser (and many other similar ones) used pre-trained word embeddings, the contextual part was learned only from the supervised treebank data. However, by using the pre-train–fine-tune paradigm of massive language models, we can directly learn a contextualized representation of words from large unannotated corpora, and later fine-tune it to the dependency parsing. The incorporation of a large contextualized language model pre-trained on a massive amount of unlabeled Finnish data, the FinBERT model in our case, improved the dependency parsing performance by several percent points, reaching the 92–94% range. Additionally, in Paper IV we showed that for several smaller languages the Google’s multilingual language model (mBERT) was not sufficient, and the parsing performance can be substantially improved with dedicated

language-specific models, the effect being especially strong on Finnish.

When considering the entire parsing pipeline, rather than solely focusing on dependency parsing, our primary emphasis was on lemmatization. Lemmatization presents interesting challenges due to Finnish inflective morphology, yet it is a sub-task that often receives insufficient attention. In RQ3, we asked *what is the most effective approach to developing a machine-learned, context-aware lemmatizer, and how would its performance compare to hand-crafted grammatical rules*. In Paper II, we demonstrated that state-of-the-art results across the UD treebanks can be attained using a sequence-to-sequence generation model. We account for the contextual nature of lemmatization by conditioning the lemma generation on the morphosyntactic features, which we demonstrated to produce a compact contextual representation that disambiguates most of the ambiguous wordforms. The approach is completely data-driven, and can be applied as-is to all languages with reasonable size of manually annotated training data. Therefore, this approach gives us a full flexibility without a need to do any language-dependent adjustments to the model. When compared to traditional rule-based methods, we gain two major advantages: 1) there is no need to craft the rules separately for each language, and 2) the model’s ability to generalize and produce output also to previously unseen words.

To conclude the methodologically oriented RQ2 and RQ3, most state-of-the-art methods are language-agnostic in a way that they use a same model architecture regardless of the language as well as rely on automatically learned feature representations. Therefore, to create a robust parsing pipeline for a language, there may not be a need to tailor the model architecture or its feature representation exclusively for that language anymore. Instead, in many cases the key to success lies in comprehending the available resources and deploying them effectively.

The parsing experiments presented in this thesis primarily focused on in-domain parsing, where the parser was evaluated on a test set sampled from its training data. For future work we leave the question how robust the Finnish parser is if applied to domains or text registers substantially deviating from its training data. Although the Finnish-TDT corpus is composed of a variety of text sources, it predominantly represents standard written language, and consequently, the parser’s performance e.g. on strong dialect or spoken language may substantially decrease. In Kanerva and Ginter (2022) we made initial contributions towards out-of-domain parsing by providing an evaluation corpus for Finnish syntactic analysis including five target domains absent from the original treebank. We also carried out preliminary parsing experiments indicating the parser being quite robust on some of the new domains (web, discussion forum), while the performance drastically dropping when really pushed into its limits (poems, clinical, tweets). In order to train a parser for domains differing from general written Finnish, a combination of in-domain data knowledge and technical skills will likely create an optimal outcome.

The achievement of (at least near) human-level performance in the syntactic pars-

ing of general Finnish brings up the hope of the large pre-trained language models genuinely comprehending language, rather than merely relying on simple surface cues. However, datasets designed to measure semantic comprehension in Finnish have been non-existent, or very scarce at the best. Consequently, our second set of research questions (RQ4 and RQ5) were centered around this theme, leading to the development of the Turku Paraphrase Corpus in Paper V. To prevent bias towards simpler and shorter paraphrase examples, which we hypothesized would be more easily recognizable using automatic paraphrase candidate extraction, we set out to study whether *the creation of a large-scale paraphrase corpus can be efficiently accomplished by manually selecting examples, thereby mitigating bias towards shorter and simpler examples (RQ4)* and whether *the resulting corpus exhibit greater diversity in terms of example length and complexity compared to corpora where candidates are automatically generated (RQ5)*.

In Paper V, we introduced a novel concept of manual paraphrase extraction from two related text documents, and explored the utility of several different data sources likely to contain naturally occurring paraphrases (alternative translations, related news articles, similar discussion forum messages, and exam answers). The different source materials exhibited diverse extraction statistics with efficiency measures revealing alternative translations as the most time-efficient. This can be attributed to translations typically adhering closely to the content of the original document, thus naturally containing extensive paraphrasing. Furthermore, content in alternative translations usually maintains the order of the original document, accelerating the extraction process by reducing the need to scroll to find corresponding segments, a convenience often absent in e.g. related news articles where the order of the information can vary significantly. In total, six annotators spent 30 person-months for the corpus construction including paraphrase extraction, label annotation as well as other related tasks such as guideline documentation. This resulted in a corpus of more than 100,000 paraphrase pairs for Finnish, which at the time of publication, was one of the largest manually annotated paraphrase corpora available for any language. Moreover, the manual extraction provided an opportunity to distribute the collected paraphrase pairs in their natural document context, making the Turku Paraphrase Corpus the first paraphrase dataset suitable for studying paraphrasing in document context.

To assess whether the resulting corpus exhibits greater diversity compared to those build using automatic candidate extraction, we evaluated the paraphrase pairs in terms of length distribution and lexical variability, making comparisons with other related works where possible. We demonstrated our corpus contains not only shorter paraphrases but also a significant proportion of longer ones while the corpora compared to leaned more towards short pairs. Additionally, we illustrated that the flexibility of manual extraction —allowing selections beyond sentence boundaries— yielded a significant number of paraphrase pairs that were not strictly sentence-to-sentence, thereby further enhancing the corpus’s variability. In measuring lexical

variance, we demonstrated that our corpus comprises a higher percentage of paraphrases with lower lexical similarity. These findings indeed validate our hypothesis that manual paraphrase extraction can yield a corpus with notably longer and less lexically overlapping pairs than what is attainable through automated candidate selection, creating a more challenging dataset, suitable, for example, for evaluating various language understanding models.

In this thesis, we presented baseline results for various paraphrase models, trained and evaluated on the Turku Paraphrase Corpus. However, a more comprehensive investigation of pre-trained language models in paraphrase modelling is reserved for future work. In the future, we plan to study the capabilities of diverse paraphrase models more closely, with a particular focus on understanding the kinds of phenomena the current models can and cannot capture. We aim to progress beyond binary classification like paraphrase-or-not, and seek to explore deeper into understanding why a certain text is considered a paraphrase and the rationale behind the model's predictions.

Recent advances in instruction fine-tuned generative models, such as GPT-4 (OpenAI, 2023), have had a significant impact on the field of Natural Language Processing. Not that long ago, the preferred method for addressing most NLP tasks involved fine-tuning pre-trained language models using supervised training data. However, the models like GPT-4 with their impressive generalization capabilities are emerging as genuine competitors to this approach, encouraging the idea of developing one universal model, interactable through human language. However, these models come with significant restrictions, as the most advanced ones are currently company owned, accessible only through API interfaces. This limits their applicability, especially in situations where data privacy is an issue which cannot be compromised. The limited availability of resources poses substantial challenges for the academic community in developing similar, but openly accessible models for various languages. Nevertheless, there are numerous initiatives ongoing to aggregate the necessary resources to train models with similar capabilities, such as BLOOM<sup>2</sup>, GPT-SW3 (Ekgren et al., 2023), or Finnish GPT-3 (Luukkonen et al., 2023).

Although these models exhibit impressive zero-shot performance across various tasks, potentially bypassing the need for task-specific training data, they still require supervised data during the instruction fine-tuning phase. Many existing datasets can be converted into valuable examples for this purpose. To create a versatile model capable of executing a wide variety of distinct tasks, instruction fine-tuning data from tasks that are diverse in nature is likely beneficial. We anticipate that the datasets developed as part of this thesis can contribute to this effort. At the very least, they can serve to evaluate the model's ability to analyze and understand the structure and meaning of the Finnish language.

---

<sup>2</sup><https://huggingface.co/bigscience/bloom>

# List of References

- Alaa Saleh Altheneyan and Mohamed El Bachir Menai. Evaluation of state-of-the-art paraphrase identification and its application to automatic plagiarism detection. *International Journal of Pattern Recognition and Artificial Intelligence*, 34, 2019.
- Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. Limsi@conll'17: Ud shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 163–173, Vancouver, Canada, August 2017. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/K/K17/K17-3017.pdf>.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL <https://aclanthology.org/W13-2322>.
- Elisa Bassignana, Filip Ginter, Sampo Pyysalo, Rob van der Goot, and Barbara Plank. Silver syntax pre-training for cross-domain relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6984–6993, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.436. URL <https://aclanthology.org/2023.findings-acl.436>.
- Toms Bergmanis and Sharon Goldwater. Context sensitive neural lemmatization with Lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1391–1400, New Orleans, Louisiana, 2018. Association for Computational Linguistics.
- Rahul Bhagat and Eduard Hovy. What is a paraphrase? *Computational Linguistics*, 39(3):463–472, 09 2013. ISSN 0891-2017. doi: 10.1162/COLI\_a\_00166. URL [https://doi.org/10.1162/COLI\\_a\\_00166](https://doi.org/10.1162/COLI_a_00166).
- Bernd Bohnet. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)*, pages 89–97, 2010.
- Bernd Bohnet, Joakim Nivre, Igor Boguslavsky, Richárd Farkas, Filip Ginter, and Jan Hajič. Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics*, 1:415–428, 2013.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X.
- Gosse Bouma, Djamé Seddah, and Daniel Zeman. Overview of the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies. In Gosse Bouma, Yuji Matsumoto, Stephan Oepen, Kenji Sagae, Djamé Seddah, Weiwei Sun, Anders Søgaard, Reut Tsarfaty, and Dan Zeman, editors, *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, Seattle, US, July 2020a. Association for Computational Linguistics. ISBN 978-1-952148-11-8.
- Gosse Bouma, Djamé Seddah, and Daniel Zeman. Overview of the IWPT 2020 shared task on parsing into enhanced Universal Dependencies. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 151–161, Online, July 2020b. Association for Computational Linguistics.

- tics. doi: 10.18653/v1/2020.iwpt-1.16. URL <https://www.aclweb.org/anthology/2020.iwpt-1.16>.
- Gosse Bouma, Djamé Seddah, and Daniel Zeman. From raw text to enhanced Universal Dependencies: The parsing shared task at IWPT 2021. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 146–157, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.iwpt-1.15. URL <https://aclanthology.org/2021.iwpt-1.15>.
- Sabine Buchholz and Erwin Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City, June 2006. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/W06-2920>.
- Abhisek Chakrabarty, Onkar Arun Pandit, and Utpal Garain. Context sensitive lemmatization using two successive bidirectional gated recurrent networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1481–1491, Vancouver, Canada, 2017. Association for Computational Linguistics.
- Li-Hsin Chang, Sampo Pyysalo, Jenna Kanerva, and Filip Ginter. Quantitative evaluation of alternative translations in a corpus of highly dissimilar Finnish paraphrases. In *Proceedings for the First Workshop on Modelling Translation: Translatology in the Digital Age, 2021a*.
- Li-Hsin Chang, Iiro Rastas, Sampo Pyysalo, and Filip Ginter. Deep learning for sentence clustering in essay grading support. In *Proceedings of the 14th International Conference on Educational Data Mining (EDM 2021)*, 2021b.
- Wanxiang Che, Jiang Guo, Yuxuan Wang, Bo Zheng, Huaipeng Zhao, Yang Liu, Dechuan Teng, and Ting Liu. The hit-scir system for end-to-end parsing of universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 52–62, Vancouver, Canada, August 2017. Association for Computational Linguistics.
- Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium, October 2018a. Association for Computational Linguistics. doi: 10.18653/v1/K18-2005. URL <https://aclanthology.org/K18-2005>.
- Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium, October 2018b. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/K18-2005>.
- Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1082. URL <https://aclanthology.org/D14-1082>.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. URL <https://openreview.net/pdf?id=r1xMH1BtvB>.
- Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL <https://aclanthology.org/2020.acl-main.747>.



- Mathias Creutz. Open Subtitles paraphrase corpus for six languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Technical report, Stanford University, 2008a.
- Marie-Catherine De Marneffe and Christopher D Manning. The stanford typed dependencies representation. In *Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*, pages 1–8, 2008b.
- Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. Universal Stanford dependencies: A cross-linguistic typology. In *LREC*, volume 14, pages 4585–4592, 2014.
- Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, 07 2021. ISSN 0891-2017. doi: 10.1162/coli\_a\_00402. URL [https://doi.org/10.1162/coli\\_a\\_00402](https://doi.org/10.1162/coli_a_00402).
- Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. BERTje: A Dutch BERT Model. *arXiv preprint arXiv:1912.09582*, 2019. URL <http://arxiv.org/abs/1912.09582>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>.
- William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP 2005)*, 2005.
- Qingxiu Dong, Xiaojun Wan, and Yue Cao. ParaSCI: A large scientific paraphrase dataset for longer paraphrase generation. *arXiv preprint arXiv:2101.08382*, 2021.
- Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2017.
- Timothy Dozat, Peng Qi, and Christopher D Manning. Stanford’s graph-based neural dependency parser at the conll 2017 shared task. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, 2017.
- Ariel Ekgren, Amaru Cuba Gyllensten, Felix Stollenwerk, Joey Öhman, Tim Isbister, Evangelia Gogoulou, Fredrik Carlsson, Alice Heiman, Judit Casademont, and Magnus Sahlgren. Gpt-sw3: An autoregressive language model for the nordic languages. *arXiv preprint arXiv:2305.12987*, 2023.
- J.R. Firth. A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis*, pages 1–32, 1957.
- Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-2103. URL <https://aclanthology.org/P18-2103>.
- Darina Gold, Venelin Kovatchev, and Torsten Zesch. Annotating and analyzing the interactions between meaning relations. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 26–36, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4004. URL <https://aclanthology.org/W19-4004>.
- Auli Hakulinen, Maria Vilkuna, Riitta Korhonen, Vesa Koivisto, Tarja Riitta Heinonen, and Irja Alho. *Iso suomen kielioppi / Grammar of Finnish*. Suomalaisen kirjallisuuden seura, 2014.
- Z. Harris. Distributional structure. *Word*, 10(23):146–162, 1954.
- Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. Building the essential resources for Finnish: the Turku Dependency Treebank. *Language Resources and Evaluation*, 48:493–531, 2014. ISSN

- 1574-020X. doi: 10.1007/s10579-013-9244-1. URL <http://dx.doi.org/10.1007/s10579-013-9244-1>. Open access.
- Katri Haverinen, Jenna Kanerva, Samuel Kohonen, Anna Missilä, Stina Ojala, Timo Viljanen, Veronika Laippala, and Filip Ginter. The Finnish Proposition Bank. *Language Resources and Evaluation*, 49(4):907–926, 2015. doi: 10.1007/s10579-015-9310-y. URL <http://dx.doi.org/10.1007/s10579-015-9310-y>.
- Yun He, Zhuoer Wang, Yin Zhang, Ruihong Huang, and James Caverlee. PARADE: A new dataset for paraphrase identification requiring computer science domain knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7572–7582, 2020.
- Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL <https://aclanthology.org/P18-1031>.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.
- Moamen Ibrahim, Matti Eteläperä, Sercan Turkmen, Mina Maged, Mourad Oussalah, and Jouko Miettunen. Mining health discussions on Suomi24. In *Proceedings of 2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom)*, 2019. doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00232.
- Ilmari Ivaska and Silvia Bernardini. Constrained language use in Finnish: A corpus-driven approach. *Nordic Journal of Linguistics*, 43(1):33–57, 2020. doi: 10.1017/S0332586520000013.
- Maciej Janicki, Eetu Mäkelä, Anu Koivunen, Antti Kanner, Auli Harju, Julius Hokkanen, and Olli Seuri. A workflow for integrating close reading and automated text annotation. In *Post-Proceedings of the 5th Conference Digital Humanities in the Nordic Countries (DHN)*, 2020.
- Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition)*. Prentice Hall, Inc., Upper Saddle River, NJ, USA, 2009. ISBN 0131873210.
- Sylvain Kahane, Marine Courtin, and Kim Gerdes. Multi-word annotation in syntactic treebanks-propositions for universal dependencies. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 181–189, 2017.
- Jenna Kanerva and Filip Ginter. Out-of-domain evaluation of Finnish dependency parsing. In *Proceedings of the 13th International Conference on Language Resources and Evaluation (LREC’22)*, pages 1114–1124, 2022. URL <http://www.lrec-conf.org/proceedings/lrec2022/pdf/2022.lrec-1.120.pdf>.
- Jenna Kanerva, Matti Luotolahti, Veronika Laippala, and Filip Ginter. Syntactic N-gram collection from a large-scale corpus of internet Finnish. In *Proceedings of the Sixth International Conference Baltic HLT 2014*, pages 184–191. IOS Press, 2014. doi: 10.3233/978-1-61499-442-8-184. URL <http://ebooks.iospress.nl/volumearticle/38025>.
- Jenna Kanerva, Samuel Rönnqvist, Riina Kekki, Tapio Salakoski, and Filip Ginter. Template-free data-to-text generation of Finnish sports news. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa’19)*, 2019.
- Jenna Kanerva, Hanna Kitti, Li-Hsin Chang, Teemu Vahtola, Mathias Creutz, and Filip Ginter. Semantic search as extractive paraphrase span detection. *Language Resources and Evaluation*, 2024. URL <https://link.springer.com/article/10.1007/s10579-023-09715-7>.
- Fred Karlsson. *Finnish: An Essential Grammar (3rd ed.)*. Routledge, 2015. doi: <https://doi-org.ezproxy.utu.fi/10.4324/9781315743233>.
- Lauri Karttunen and Kenneth R Beesley. *Two-level rule compiler*. Xerox Corporation. Palo Alto Research Center, 1992.

- Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327, 2016. doi: 10.1162/tacl\_a\_00101. URL <https://aclanthology.org/Q16-1023>.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of the 55th annual meeting of the Association for Computational Linguistics (ACL’17)*, Vancouver, Canada, 2017. Association for Computational Linguistics.
- Dan Kondratyuk and Milan Straka. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China, 2019. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D19-1279>.
- Kimmo Koskenniemi. A general computational model for word-form recognition and production. In *Proceedings of the 10th international conference on Computational Linguistics*, pages 178–181, USA, 1984. Association for Computational Linguistics.
- Artur Kulmizev, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. Deep contextualized word embeddings in transition-based and graph-based dependency parsing - a tale of two parsers revisited. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2755–2768, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1277. URL <https://aclanthology.org/D19-1277>.
- Yuri Kuratov and Mikhail Arkhipov. Adaptation of deep bidirectional multilingual transformers for Russian language. *arXiv preprint arXiv:1905.07213*, 2019.
- Karoliina Kuusinen. Ulkomaisten yliopistojen suomenoppijoiden leksikaalisen diversiteetin kehittymisen intensiivikurssilla Suomessa. Master’s thesis, University of Turku, 2021.
- Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. A continuously growing dataset of sentential paraphrases. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1126. URL <https://aclanthology.org/D17-1126>.
- Krister Lindén. A probabilistic model for guessing base forms of new words by analogy. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 106–116. Springer, 2008.
- Krister Lindén, Miikka Silfverberg, and Tommi Pirinen. Hfst tools for morphology—an efficient open-source package for construction of morphological analyzers. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 28–47. Springer, 2009.
- Juhani Luotolahti, Jenna Kanerva, Veronika Laippala, Sampo Pyysalo, and Filip Ginter. Towards universal web parsebanks. In *Proceedings of the International Conference on Dependency Linguistics (Depling’15)*, pages 211–220. Uppsala University, 2015.
- Risto Luukkonen, Ville Komulainen, Jouni Luoma, Anni Eskelinen, Jenna Kanerva, Hanna-Mari Kupari, Filip Ginter, Veronika Laippala, Niklas Muennighoff, Aleksandra Piktus, Thomas Wang, Nouamane Tazi, Teven Scao, Thomas Wolf, Osmo Suominen, Samuli Sairanen, Mikko Merioksa, Jyrki Heinonen, Aija Vahtola, Samuel Antao, and Sampo Pyysalo. FinGPT: Large generative models for a small language. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2710–2726, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.164. URL <https://aclanthology.org/2023.emnlp-main.164>.
- Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd*

- annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.
- Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamel Seddah, and Benoît Sagot. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online, July 2020. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/2020.acl-main.645>.
- Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1334. URL <https://aclanthology.org/P19-1334>.
- Ryan McDonald and Fernando Pereira. Online learning of approximate dependency parsing algorithms. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 81–88, Trento, Italy, April 2006. Association for Computational Linguistics. URL <https://aclanthology.org/E06-1011>.
- Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics. URL <https://aclanthology.org/H05-1066>.
- Ryan McDonald, Joakim Nivre, Yvonne Quirnbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, 2013.
- Ramtin Mehdizadeh Seraj, Maryam Siahbani, and Anoop Sarkar. Improving statistical machine translation with a multilingual paraphrase database. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1379–1390, 2015.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at International Conference on Learning Representations (ICLR)*, 2013.
- Thomas Müller, Helmut Schmid, and Hinrich Schütze. Efficient higher-order crfs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, 2013.
- Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. Joint lemmatization and morphological tagging with Lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274, Lisbon, Portugal, 2015. Association for Computational Linguistics.
- Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 2021.
- Malvina Nissim, Lasha Abzianidze, Kilian Evang, Rob van der Goot, Hessel Haagsma, Barbara Plank, and Martijn Wieling. Last words: Sharing is caring: The future of shared tasks. *Computational Linguistics*, 43(4):897–904, December 2017. doi: 10.1162/COLI\_a\_00304. URL <https://aclanthology.org/J17-4007>.
- Joakim Nivre. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://aclanthology.org/W04-0308>.
- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural*

- Language Learning (EMNLP-CoNLL)*, pages 915–932, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D07-1096>.
- Joakim Nivre, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Yoav Goldberg, Jan Hajič, Jenna Kanerva, Veronika Laipala, Alessandro Lenci, Teresa Lynn, Christopher Manning, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Maria Simi, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. Universal dependencies 1.0, 2015. URL <http://hdl.handle.net/11234/1-1464>. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, 2016.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, and Sebastian Schuster. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of 12th Conference on Language Resources and Evaluation LREC'2020*, 2020.
- Stephan Open, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. SemEval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72. Association for Computational Linguistics, 2014.
- Stephan Open, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. SemEval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926. Association for Computational Linguistics, 2015.
- OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th Conference on Neural Information Processing System, 2022*.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106, 2005.
- Carla Parra Escartín, Wessel Reijers, Teresa Lynn, Joss Moorkens, Andy Way, and Chao-Hong Liu. Ethical considerations in NLP shared tasks. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 66–73, Valencia, Spain, April 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-1608. URL <https://aclanthology.org/W17-1608>.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL <http://www.aclweb.org/anthology/D14-1162>.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL <https://aclanthology.org/N18-1202>.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, pages 2089–2096, 2012.

- Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online, October 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.7. URL <https://aclanthology.org/2020.emnlp-demos.7>.
- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.617. URL <https://aclanthology.org/2020.emnlp-main.617>.
- Tommi Pirinen. Suomen kielen äärellistilainen automaattinen morfologinen jäsennin avoimen lähdekoodin resurssin. *Master's Thesis, University of Helsinki*, 2008.
- Lidia Pivovarova, Ekaterina Pronoza, Elena Yagunova, and Anton Pronoza. Paraphraser: Russian paraphrase corpus and shared task. In Andrey Filchenkov, Lidia Pivovarova, and Jan Žižka, editors, *Artificial Intelligence and Natural Language*, pages 211–225. Springer International Publishing, 2018. ISBN 978-3-319-71746-3.
- Sampo Pyysalo, Jenna Kanerva, Antti Virtanen, and Filip Ginter. WikiBERT models: Deep transfer learning for many languages. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021)*, 2021. URL <https://www.aclweb.org/anthology/2021.nodalida-main.1.pdf>.
- Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. Universal Dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium, October 2018. Association for Computational Linguistics.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL <http://jmlr.org/papers/v21/20-074.html>.
- Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL <https://aclanthology.org/D19-1410>.
- Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.365. URL <https://aclanthology.org/2020.emnlp-main.365>.
- Samuel Rönnqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. Is multilingual BERT fluent in language generation? In *Proceedings of the 1st NLPL Workshop on Deep Learning for Natural Language Processing*, 2019.
- Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In Anoop Sarkar and Michael Strube, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:*

- Tutorials*, pages 15–18, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-5004. URL <https://aclanthology.org/N19-5004>.
- Kenji Sagae and Jun’ichi Tsujii. Shift-reduce dependency DAG parsing. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 753–760, Manchester, UK, August 2008. Coling 2008 Organizing Committee. URL <https://aclanthology.org/C08-1095>.
- Yves Scherrer. TaPaCo: A corpus of sentential paraphrases for 73 languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6868–6873, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL <https://aclanthology.org/2020.lrec-1.848>.
- Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Gallettebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/W13-4917>.
- Olli Seuri, Riikka Era, Anu Koivunen, Maciej Janicki, Pihla Toivanen, Julius Hokkanen, and Eetu Mäkelä. Uutisvuon hallitsija: Uutismedia kiky-kamppailussa 2015–2016. *Politiikka*, 63(3), syys 2021. doi: 10.37452/politiikka.99432. URL <https://journal.fi/politiikka/article/view/99432>.
- Natalia Silveira and Christopher Manning. Does Universal Dependencies need a parsing representation? an investigation of English. In Joakim Nivre and Eva Hajičová, editors, *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 310–319, Uppsala, Sweden, August 2015. Uppsala University, Uppsala, Sweden. URL <https://aclanthology.org/W15-2134>.
- Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 82 treebanks, 34 models: Universal dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium, October 2018. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/K18-2011>.
- Sarvesh Soni and Kirk Roberts. A paraphrase generation system for EHR question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 20–29, 2019.
- Milan Straka. UDPipe 2.0 prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium, October 2018. Association for Computational Linguistics.
- Milan Straka and Jana Straková. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipes. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, August 2017. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/K/K17/K17-3009.pdf>.
- Milan Straka, Jan Hajič, Jana Straková, and Jan Hajič jr. Parsing universal dependency treebanks using neural networks and search-based oracle. In *Proceedings of Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT 14)*, December 2015.
- Milan Straka, Jan Hajič, and Jana Straková. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, 2016. European Language Resources Association. ISBN 978-2-9517408-9-1.
- Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng

- Wang, Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*, 2021.
- Masatoshi Tsuchiya. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018. European Language Resources Association (ELRA). URL <https://aclanthology.org/L18-1239>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.
- David Vilares and Carlos Gómez-Rodríguez. A non-projective greedy dependency parser with bidirectional lstms. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 152–162, Vancouver, Canada, August 2017. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/K/K17/K17-3016.pdf>.
- Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. Multilingual is not enough: Bert for Finnish, 2019.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL <https://aclanthology.org/W18-5446>.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf>.
- John Wieting and Kevin Gimpel. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1042. URL <https://aclanthology.org/P18-1042>.
- Tatu Ylönen. Wiktextextract: Wiktionary as machine-readable structured data. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1317–1325, Marseille, France, June 2022. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.140>.
- Kuan Yu, Pavel Sofroniev, Erik Schill, and Erhard Hinrichs. The parse is dark and full of errors: Universal dependency parsing with transition-based and graph-based algorithms. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 126–133, Vancouver, Canada, August 2017. Association for Computational Linguistics.
- Daniel Zeman. Reusable tagset conversion using tagset drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, volume 2008, pages 28–30, 2008.
- Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčková, Václava Kettnerová, Zdeňka Uřešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D.



- Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Drostanova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaraj, and Josie Li. CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-3001. URL <https://www.aclweb.org/anthology/K17-3001>.
- Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 1–21, 2018.
- Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Salih Furkan Akkurt, Gabrielë Aleksandravičiūtė, Ika Alfina, Avner Algom, Khalid Alnajjar, Chiara Alzetta, Erik Andersen, Lene Antonsen, Tatsuya Aoyama, Katya Aplonova, Angelina Aquino, Carolina Aragon, Glyd Aranes, Maria Jesus Aranzabe, Bilge Nas Arican, Hórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Katla Ásgeirsdóttir, Deniz Baran Aslan, Cengiz Asmazoğlu, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Mariana Avelãs, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Shabnam Behzad, Kepa Bengoetxea, Ibrahim Benli, Yifat Ben Moshe, Gözde Berk, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, António Branco, Kristina Brokaitė, Aljoscha Burchardt, Marisa Campos, Marie Candito, Bernard Caron, Gauthier Caron, Catarina Carvalheiro, Rita Carvalho, Lauren Cassidy, Maria Clara Castro, Sérgio Castro, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Liyanage Chamila, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Daniela Corbetta, Francisco Costa, Marine Courtin, Mihaela Cristescu, Ingerid Løyning Dale, Philemon Daniel, Elizabeth Davidson, Leonel Figueiredo de Alencar, Mathieu Dehouck, Martina de Laurentiis, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Adrian Doyle, Timothy Dozat, Kira Drostanova, Puneet Dwivedi, Christian Ebert, Hanne Eckhoff, Masaki Eguchi, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaz Erjavec, Farah Essaidi, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Federica Favero, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Amal Fethi, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Federica Gamba, Marcos Garcia, Moa Gärdenfors, Fabrício Ferraz Gerardi, Kim Gerdes, Luke Gessler, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mý, Na-Rae Han, Muhammad Yudistira Hanifmuti, Takahiro Harada, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava

Hlaváčová, Florinel Hociung, Petter Hohle, Marivel Huerta Mendez, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Olájídé Ishola, Artan Islamaj, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Katharine Jiang, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Ritvan Karahoga, Andre Kasen, Tolga Kayadelen, Sarveswaran Kengatharaiyer, Vaclava Kettnerova, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Kohn, Abdullatif Koksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Kose, Alexey Koshevoy, Natalia Kotsyba, Jolanta Kovalevskaite, Simon Krek, Parameswari Krishnamurthy, Sandra Kubler, Adrian Kuqi, Oğuzhan Kuyruku, Aslı Kuzgun, Sookyoung Kwak, Kris Kyle, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Le Hong, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Lauren Levine, Cheuk Ying Li, Josie Li, Keying Li, Yixuan Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Yi-Ju Jessica Lin, Krister Linden, Yang Janet Liu, Nikola Ljubešic, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Ilya Makarchuk, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Buřra Marřan, Catalina Maranduc, David Marecek, Katrin Marheinecke, Stella Markantonatou, Hector Martinez Alonso, Lorena Martın Rodrıguez, Andre Martins, Claudia Martins, Jan Mařek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonca, Tatiana Merzhevich, Niko Miekka, Aaron Miller, Karina Mischenkova, Anna Missila, Catalin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnar, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskiy, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Muurisep, Pinkey Nainwani, Mariam Nakhle, Juan Ignacio Navarro Horıacek, Anna Nedoluzhko, Gunta Neřpore-Berzkalne, Manuela Nevaci, Lng Nguyen Thi, Huyen Nguyen Thi Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Hulda Oladottir, Adedayo Oluokun, Mai Omura, Emeka Onwuegbuzia, Noam Ordan, Petya Osenova, Robert Ostling, Lilja Ovrelid, řaziye Betul Ozateř, Merve Ozelik, Arzucan Ozgur, Balkız Ozturk Bařaran, Teresa Paccosi, Alessio Palmero Aprosio, Anastasia Panova, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Giulia Pedonese, Angelika Peljak-Lapiņska, Siyao Peng, Siyao Logan Peng, Rita Pereira, Silvia Pereira, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Andrea Peverelli, Jason Phelan, Jussi Piitulainen, Yuval Pinter, Clara Pinto, Tommi A Pirinen, Emily Pitler, Magdalena Plamada, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prevost, Prokopis Prokopidis, Adam Przepiorkowski, Robert Pugh, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andreia Querido, Andriela Raabis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Joana Ramos, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Arij Riabi, Ivan Riabov, Michael Rieβler, Erika Rimkute, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eirkur Rognvaldsson, Ivan Roksandic, Mykhailo Romanenko, Rudolf Rosa, Valentin Rořca, Davide Rovati, Ben Rozonoyer, Olga Rudina, Jack Rueter, Kristjan Runarsson, Shoval Sadde, Pegah Safari, Aleksı Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžic, Stephanie Samson, Manuela Sanguinetti, Ezgi Saniyar, Dage Sarg, Marta Sartor, Mitsuya Sasaki, Baiba Saulite, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djame Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Maria Shvedova, Janine Siewert, Einar Freyr Sigursson, Joao Silva, Aline Silveira, Natalia Silveira, Sara Silveira, Maria Simi, Radu Simionescu, Katalin Simko, Maria řimkova, Haukur Barri Sımonarson, Kiril Simov, Dmitri Sitchinava, Ted Sither, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Per Erik Solberg, Barbara Sonnenhauser, Shafi Surov, Rachele Sprugnoli, Vivian Stamou, Steinhor Steingrımsson, Antonio Stella,

Abishek Stephen, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Daniel Swanson, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Mirko Tavoni, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Sara Tonelli, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sveinbjörn Hórdarson, Vilhjálmur Hörsteinsson, Sumire Uematsu, Roman Untilov, Zdeňka Uřešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Elena Vagnoni, Sowmya Vajjala, Socrates Vak, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Uliana Vedenina, Giulia Venturi, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jonathan North Washington, Maximilian Wendt, Paul Widmer, Shira Wigderson, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Wolde-mariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Yilun Zhu, Anna Zhuravleva, and Rayan Ziane. Universal dependencies 2.12, 2023. URL <http://hdl.handle.net/11234/1-5150>. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.



# Original Publications

**Sampo Pyysalo & Jenna Kanerva & Anna Missilä &  
Veronika Laippala & Filip Ginter  
Universal Dependencies for Finnish**

In Proceedings of the 20th Nordic Conference of Computational Linguistics  
(NODALIDA). 2015.

# Universal Dependencies for Finnish

Sampo Pyysalo<sup>1</sup> Jenna Kanerva<sup>1,2</sup> Anna Missilä<sup>4</sup> Veronika Laippala<sup>3,4</sup> Filip Ginter<sup>1</sup>

<sup>1</sup>Department of Information Technology <sup>2</sup>University of Turku Graduate School (UTUGS)

<sup>3</sup>Turku Institute for Advanced Studies (TIAS) <sup>4</sup>School of Languages and Translation Studies,  
University of Turku, Finland

first.last@utu.fi

## Abstract

There has been substantial recent interest in annotation schemes that can be applied consistently to many languages. Building on several recent efforts to unify morphological and syntactic annotation, the Universal Dependencies (UD) project seeks to introduce a cross-linguistically applicable part-of-speech tagset, feature inventory, and set of dependency relations as well as a large number of uniformly annotated treebanks. We present Universal Dependencies for Finnish, one of the ten languages in the recent first release of UD project treebank data. We detail the mapping of previously introduced annotation to the UD standard, describing specific challenges and their resolution. We additionally present parsing experiments comparing the performance of a state-of-the-art parser trained on a language-specific annotation schema to performance on the corresponding UD annotation. The results show improvement compared to the source annotation, indicating that the conversion is accurate and supporting the feasibility of UD as a parsing target. The introduced tools and resources are available under open licenses from <http://bionlp.utu.fi/ud-finnish.html>.

## 1 Introduction

The Universal Dependencies (UD) initiative seeks to develop cross-linguistically consistent annotation guidelines and apply them to many languages to create treebank annotations that are uniform in e.g. their theoretical basis, label sets, and structural aspects. Such resources could substantially advance cross-lingual learning, improve comparability of evaluation results, and facilitate new approaches to automatic syntactic analysis.

UD builds on the Google Universal part-of-speech (POS) tagset (Petrov et al., 2012), the Intersect interlingua of morphosyntactic features (Zeman, 2008), and Stanford Dependencies (de Marneffe et al., 2006; Tsarfaty, 2013; de Marneffe et al., 2014). In addition to the abstract annotation scheme, UD defines also a treebank storage format, CoNLL-U. A first version of UD treebank data, building on the Google Universal Dependency Treebanks (McDonald et al., 2013) and many other previously released resources (Bosco et al., 2013; Haverinen et al., 2013b), was recently released<sup>1</sup> (Nivre et al., 2015).

In this paper, we present the adaptation of the UD guidelines to Finnish and the creation of the UD Finnish treebank by conversion of the previously introduced Turku Dependency Treebank (TDT) (Haverinen et al., 2013b). We also provide a first set of experiments comparing the parsing scores of language-specific treebank annotation to that of a UD treebank, providing an evaluation of both the conversion quality and the feasibility of UD annotation as a parsing target. In a related but separate effort within the UD initiative, the FinnTreeBank 1<sup>2</sup> (ftb-1) (Voutilainen, 2011) is also being converted into the UD format. The *ftb-1* is a treebank based on all grammatical examples from the VISK<sup>3</sup> Finnish grammar reference (Hakulinen et al., 2004), and will thus complement the TDT-based UD Finnish treebank in the set of UD treebanks.

## 2 Treebank conversion

The conversion of TDT into the UD Finnish treebank was implemented following the UD specification (Nivre et al., 2014) (version 1, Oct 2014),

<sup>1</sup>Available from <http://universaldependencies.github.io/docs/>

<sup>2</sup><http://www.ling.helsinki.fi/kieliteknologia/tutkimus/treebank/sources/>

<sup>3</sup><http://scripta.kotus.fi/visk>

the Finnish grammar of Hakulinen et al. (2004) and the TDT annotation guidelines (Haverinen et al., 2013b) as the primary references. The initial stages of the work involved identifying similarities and differences between the TDT and UD annotation guidelines, adapting the general UD guidelines to Finnish, and planning the implementation of the conversion. Technically, the conversion was implemented as a pipeline of processing components, each of which consumed and produced CoNLL-U-formatted data. The following sections present the source data and primary stages of processing in detail.

## 2.1 Turku Dependency Treebank

As the source data for the conversion, we selected the most recent published distribution of TDT.<sup>4</sup> The source treebank contains 15,000 sentences (200,000 words) drawn from a variety of sources and annotated in a Finnish-specific version of the Stanford Dependencies (SD) scheme, and it has previously been demonstrated to be applicable e.g. for training broad-coverage dependency parsers for Finnish (Kanerva et al., 2014).

In addition to converting the annotation to UD standards, we also addressed a number of instances where tokenization differed from UD specifications, corrected a small number of sentence-splitting errors, and updated the lemmas to improve both treebank-internal consistency and conformance with the UD specification. We further introduced a fully manually annotated morphology layer, replacing the automatically generated morphological annotation of the initial data. This modified TDT not only serves as the basis for conversion but is also made available as a separate contribution.

## 2.2 Part-of-speech annotation

The UD specification defines 17 POS tags, and requires that all conforming treebanks use only these tags.<sup>5</sup> The TDT annotation uses a comparatively coarse-grained set of 12 POS tags, of which approximately half correspond straightforwardly to one of the 17 UD POS tags (Table 1). Several other TDT tags could be assigned the appropriate UD tag based on the value of the SUBCAT fea-

TDT	UD	TDT type
A	ADJ	adjective
Adp	ADP	adposition
Adv	ADV	adverb
C[SUBCAT=CC]	CONJ	coord. conj.
C[SUBCAT=CS]	CONJ	subord. conj.
Foreign	X	foreign word
Interj	INTJ	interjection
N[SUBCAT=Prop]	PRONP	proper noun
N[!SUBCAT=Prop]	NOUN	common noun
Num[SUBCAT=Card]	NUM	cardinal number
Num[SUBCAT=Ord]	ADJ	ordinal number
Pron	PRON or ADJ	pronoun
Punct	PUNCT or SYM	punctuation
Symb	PUNCT or SYM	symbol
V	VERB or AUX	verb

Table 1: Part-of-speech tag mapping from TDT to UD. TAG[FEATURE=VALUE] specifies a mapping that applies only in cases where a word has both the given tag and the feature value, TAG[!FEATURE=VALUE] in cases where the feature is absent or has a different value.

ture, which distinguishes e.g. coordinating conjunctions from subordinating conjunctions (CONJ and SCONJ in UD, respectively). Just four TDT tags, marking pronouns, punctuation, symbols and verbs, required further information to resolve correctly.

**Punctuation and symbols** The guidelines covering the use of the Punct and Sym tags in the TDT annotation differed to such an extent from the UD specification of PUNCT and SYM that the Punct/Sym distinction in the original treebank was ignored in creating the mapping. Instead, words assigned either of these tags in TDT were assigned UD POS based on newly implemented surface form-based heuristics, with e.g. currency symbols, mathematical operators, URLs and emoticons assigned SYM and other non-alphabetical character sequences PUNCT.

**Verbs** All verbs that can serve as auxiliaries were assigned AUX or VERB based on the presence of an aux dependency. This is the only rule concerning the morphological annotation layer that refers to the syntactic annotation. It should be noted that this rule cannot be applied deterministically in a standard syntactic analysis pipeline where morphological analysis precedes dependency analysis, but will instead require these verbs to be assigned both a VERB and AUX reading.

**Pronouns** The TDT POS tag Pron maps to PRON for UD Finnish in most cases, but pro-

<sup>4</sup>Available from <http://bionlp.utu.fi/>

<sup>5</sup>While no language-specific POS tags can thus be defined in the primary POS annotation, the CoNLL-U format allows a secondary POS tag to be assigned to each word to preserve treebank-specific information.

adjectives such as *millainen* “like-what” are analyzed as Pron in TDT but assigned to ADJ in UD Finnish following the reference grammar and the UD specification. The annotation of related cases such as pro-adverbs was already consistent with the reference resources and could thus be processed using the general mapping rules.

Finally, we note that UD Finnish excludes by design two of the UD POS tags, DET (determiner) and PART (particle). As Finnish has no true articles (Sulkala and Karjalainen, 1992) and words (primarily pronouns) that play a determiner role syntactically can be identified using the dependency annotation layer (namely, the det relation), we opted not to apply DET in UD Finnish annotation. Similarly, although various words have been categorized as particles in different descriptions of Finnish, the reference grammar (Hakulinen et al., 2004) does not assign any Finnish words to the category covered by PART in the UD specification. This POS tag is correspondingly excluded from use in UD Finnish.

### 2.3 Morphological features

The UD specification defines a set of 17 widely attested morphological features such as Case, Person, Number, Voice and Mood. However, by contrast to the POS tag annotation, the specification allows conforming treebanks to introduce language-specific features that are not included in this universal inventory, suggesting that such features be drawn when possible from the extended Intersect compilation of morphological feature names and labels (Zeman, 2008).

The morphological annotation of TDT draws directly on the rich features provided by the OMorFi morphological analyzer (Pirinen, 2008), and many of the generally applicable UD features can be generated by direct mapping from TDT POS tags and features (Table 2). For brevity, we refer to UD documentation for descriptions of UD standard features, focusing in the following on UD Finnish features not among the basic 17.

To minimize information loss from the conversion, we made liberal use of the possibility to introduce language-specific features to mark aspects of the TDT morphological annotation that were not captured by the basic 17 UD features. We aimed to primarily apply extended Intersect features, drawing from this inventory the features Abbr (abbreviation or acronym), Style (collo-

TDT	UD
CASE	Case
CLIT	Clitic
CMP	Degree
DRV	Derivation
INF	InfForm and VerbForm=Inf
MOOD	Mood
NEG=ConNeg	Connegative=Yes
OTHER=Coll	Style=Coll
OTHER=Arch	Style=Arch
OTHER=Err	Typo=Yes
PCP	PartForm and VerbForm=Part
POSS	Person[psor] and Number[psor]
V[SUBCAT=Neg]	Negative=Yes
SUBCAT=Pfx	-
Pron[SUBCAT]	PronType or Reflex
Adp[SUBCAT]	AdpType
SUBCAT=Card Ord	NumType
NUM	Number
TENSE	Tense
VOICE	Voice
PRS	Person and Number
ABBR	Abbr
ACRO	Abbr
not INF and not PCP	VerbForm=Fin
FOREIGN[...]	Foreign

Table 2: Morphological feature mapping. FEATURE denotes a mapping that applies for all features with the given name, FEATURE=VALUE for a specific name-value pair, and TAG[FEATURE=VALUE] also for a specific POS tag. Person[psor] and Number[psor] are layered UD features for Person and Number of possessor, respectively.

quial or archaic style), Typo (typographic error), Foreign (foreign word or script) and AdpType (adposition type: pre- or postposition). Finally, we added features to capture aspects of TDT annotation that did not have representation in Intersect: InfForm (differentiates between Finnish infinitives), PartForm (similar for participles), Connegative (verb in connegative form) and Clitic and Derivation, identifying steps in the morphological derivation and modification processes to create the wordform.

While the great majority of UD Finnish features could be deterministically generated by reference only to the TDT POS tag and features, there were a few cases that required more complex heuristics to meet UD requirements. For example, the value of the Person feature is assigned to personal pronouns based on a lemma lookup table as OMorFi does not generate it, and the value of the Foreign value is assigned based on comparison of characters in the surface form against Unicode script tables.



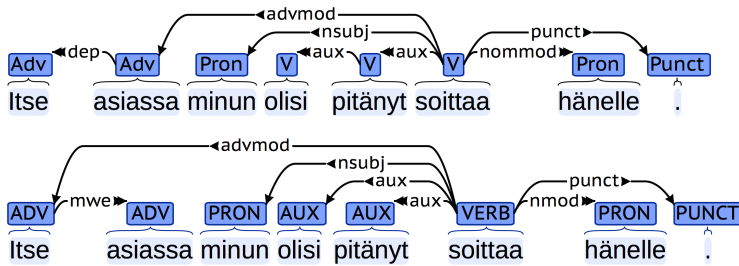


Figure 1: Top: TDT-style syntax and part-of-speech annotation for a Finnish sentence. Bottom: The same sentence converted to the UD Finnish scheme. Analyses visualized using BRAT (Stenetorp et al., 2012).

## 2.4 Dependency annotation

UD defines a set of 40 broadly applicable dependency relations, further allowing language-specific subtypes of these to be defined to meet the needs of specific resources. Unlike the fairly straightforward mappings for morphological annotations, the conversion from TDT dependency annotation to UD often required not only relabeling types, but also changes to the tree structure. This mapping is summarized in Table 3 and presented in detail below.

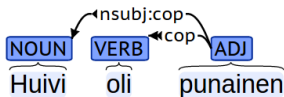


Figure 2: Annotation of *Huiivi oli punainen* “The scarf was red”.

The UD syntactic annotation is based on the universal Stanford Dependencies (SD) scheme (de Marneffe et al., 2014). One of the key properties of these schemes is that they emphasize direct relations between content words, treating function words as dependents of content words rather than as their heads. For example, this leads to a structure where a copula subject is attached directly to the predicative with the copular verb also becoming a dependent of the predicative (Figure 2). Furthermore, function words can only have a very limited set of dependents, with strong preference given to attachment of function words to content words rather than to other function words. This will tend to produce relatively flat tree structures.

The UD emphasis on content words is not universally shared with other dependency annotation schemes, many of which mediate connections between content words through function words.

However, TDT is originally annotated using a language-specific variant of the SD scheme, and thus already applies an annotation scheme with predicatives as heads in copular expressions and content-word heads in prepositional phrases. The conversion of the syntactic annotation to UD thus involved fewer challenges than might be encountered for other treebanks.

During the conversion, relatively few structural reconfigurations were required. In the original TDT annotation, function words were allowed to have dependents of their own, permitting e.g. chains of auxiliary verbs (see Figure 1). These modifiers were reattached to the upper-level content words. Additionally, multi-word expressions and names were annotated with head-final structures in TDT, but UD specifies head-initial annotation for all expressions that do not have internal structure of their own. For UD Finnish, multi-word expressions were revised to follow the UD head-initial approach. However, the head-final structure was kept for names. This decision reflects the fact that in Finnish multi-word names, only the last word carries the morphological inflections, providing evidence that it is the head of the phrase. By contrast, fixed multi-word expressions (UD *mwe*) do not typically inflect, and thus do not provide sufficient cause to diverge from the UD guideline of head-initial annotation.

One problematic issue arose from the fact that UD makes a systematic distinction between core arguments and other modifiers, which are only partly distinguished in TDT annotation. For example, participial modifiers of predicates, which usually include also secondary predication, were annotated simply as participial modifiers in TDT, while in UD these are seen as clausal dependents and a distinction must thus be made between com-

<b>Unchanged types</b>
advcl, amod, appos, aux, auxpass, cc, conj, cop, csubj, det, dobj, mark, name, nsubj, neg, root, parataxis, xcomp
<b>Simple mapping</b>
acompl → xcomp, adpos → case, compar → advcl, comparator → mark, complm → mark, csubj-cop → csubj:cop, gobj → nmod:gobj, gsubj → nmod:gsubj, icomp → xcomp:ds, infmod → acl, intj → discourse, nommod-own → nmod:own, nsubj-cop → nsubj:cop, num → nummod, number → compound, poss → nmod:poss, preconj → cc:preconj, prt → compound:prt, quantmod → advmod, rcmmod → acl:relcl, voc → vocative, xsubj → nsubj, xsubj-cop → nsubj:cop
<b>More complex mapping</b>
advmod → advmod, cc, mark
ccomp → ccomp, xcomp:ds
dep → dep, mwe
nommod → nmod, xcomp, xcomp:ds
nn → compound:nn, goeswith
partmod → acl, advcl, ccomp, xcomp, xcomp:ds
punct → discourse, punct
∅ → remnant
<b>Unmapped TDT types (removed)</b>
ellipsis, rel
<b>Unused UD types</b>
csubjpass, dislocated, foreign, expl, iobj, list, nsubjpass, reparandum

Table 3: Dependency type mapping from TDT to UD Finnish.

plements and adjuncts. To implement the conversion for cases like these, we made reference to the manually annotated predicate-argument structures of the Finnish Propbank (Haverinen et al., 2013a). Since the Finnish Propbank and the Turku Dependency Treebank are built on top of the same texts, we had access to semantic information where each argument is marked to identify whether it serves as a core argument or a modifier.

In some cases the original TDT annotation is more fine-grained than the relation types defined in the UD guidelines. We use two approaches to resolve this issue in UD Finnish. First, most of the more specific dependency types not defined in UD are simply dropped from UD Finnish, replacing occurrences of the types with their more general UD types. This is done in particular for TDT types that are not specific to Finnish and encode distinctions not targeted in UD syntactic relations, such as the difference between finite and non-finite clauses (cf. SD partmod and infmod). However, some fine-grained dependencies were defined in the TDT variant of the SD scheme to capture properties that are unique or especially important to the Finnish language. We introduce some of these relations also in UD Finnish as subtypes of UD relations. This allows us to preserve the information while allowing a fully comparable UD analysis to be generated by simply replacing detailed types with those that they are subtypes of. For example, Finnish does not have a specific verb express-

ing ownership (such as *to have* in English), and typically the verb *olla* “to be” is used instead with the owner expressed with a nominal modifier. The surface forms of possessive clauses and existential clauses are similar (*Minulla on koira* “I have a dog”, lit. *At me is a dog* and *Pihalla on koira* “These is a dog in the yard”), and using the standard nominal modifier type nmod for both would fail to distinguish these constructions. Thus, UD Finnish carries over the original TDT distinction and defines a language-specific subtype nmod:own to address this issue. nmod:own can then trivially be mapped to nmod when the distinction is not required.

The total number of dependency relation types defined in UD Finnish is 43, consisting of 32 universal relations and 11 language-specific subtypes. In the original TDT annotation, 46 dependency types are used, with an additional 4 types to mark non-tree structures used in the second annotation layer of TDT. In UD Finnish, the second annotation layer does not expand the set of dependency types. Although not currently formalized in UD, the *extended* layer of annotation from TDT (Haverinen et al., 2013b) was converted as well and is included in the UD version of TDT. This extended TDT layer includes (1) conjunct propagation, where dependencies of the head of a coordination structure are propagated where applicable also to the other coordinated elements, (2) external subjects (xsubj) of open clausal complements,

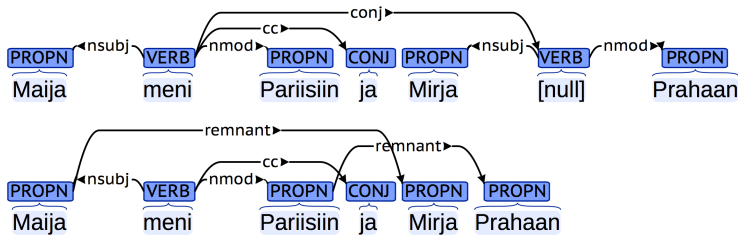


Figure 3: TDT-style (top) and UD-style (bottom) analysis for the sentence *Maija meni Pariisiin ja Mirja Prahaan* “Maija went to Paris and Mirja to Prague”.

(3) name dependencies marking named entities spanning several words and having some internal syntactic structure, (4) dependencies marking the syntactic function of relativizers, and (5) the ellipsis dependency marking constructions involving ellipsis. Of these, conjunct propagation is converted using the same rules as the base syntax dependencies, external subjects are renamed to the standard subject relation *nsubj* or the language-specific *nsubj:cop* copula subject relation, the name dependencies are preserved, dependencies marking the syntactic function of relativizers are converted and placed into the base layer, replacing the *rel* dependency type (which is eliminated) and *ellipsis* dependencies are removed together with the *null* nodes they marked. (We refer to the UD Finnish documentation for further details.)

### 2.4.1 Implementation

While POS tags and morphological features could be mapped with rules affecting a single word and only referencing properties of that word, the dependency annotation mapping requires changes to the tree structure and the ability to refer to a wider syntactic context in mapping rules. The conversion is implemented using the *dep2dep* tool which allows rules that produce dependencies in the output tree based on an input tree context that can be specified in considerable detail: it can match subtree structures, specify negations (e.g. *does not have a property, dependent, or subtree*), refer to the morphological layer, the linear order of tokens, and to additional meta-data such as PropBank argument roles. The tool is implemented as a compiler that converts the source expressions into predicates in Prolog, which is then used to apply the rules.

As an illustration, consider the rule below, which specifies that an *advcl* UD dependency is to be produced between a verb and its participial

modifier *partmod* in the transitive case, providing that the participle is not a core argument of the verb in the PropBank.

```
[v p ('advcl')] : [
  @[-"POS_V" p-"CASE=Tra" ("partmod")]
  ! [v p ("Arg_*")]
]
```

In total, the conversion consists of 116 such rules, of which 22 are simple direct dependency renamings, and the remaining refer to a broader context. We note that these rules did not aim to be universal or exhaustive: a small number of dependencies, on the order of 250, were not covered by the rules and were edited manually upon conversion. This was more efficient than writing rules that only apply to generate very few or only single dependencies.

### 2.4.2 Null tokens

In many situations sentences can be incomplete and elements obvious from the context can be omitted. In *gapping*, an elliptic sentence element is omitted to avoid unnecessary repetition, whereas in *sentence fragments* the main predicate is absent. The analysis of fragments and sentences including gapping is difficult, and many different approaches have been proposed. In TDT the omitted token, most commonly a verb, is replaced with a *null* token, which is given a full morphological analysis and which acts as a normal token in the syntactic analysis.

UD takes a different approach to analyzing omitted sentence elements. UD aims in general to avoid representing things that are absent, and does not define a way to introduce null tokens. Instead, for example to address coordination with ellipsis, UD introduces a special dependency type *remnant*. Thus, e.g. *Maija meni Pariisiin ja Mirja Prahaan* “Maija went to Paris and Mirja to Prague” is analysed with an empty token representing *meni* “went” in the second constituent in

Language	Tokens	Source treebank
Czech	1,506,490	Prague Dependency Treebank 3.0 (PDT) (Bejček et al., 2012)
Spanish	432,651	Universal Dependency Treebank v2.0 (UDT) (McDonald et al., 2013)
French	400,620	Universal Dependency Treebank v2.0 (UDT) (McDonald et al., 2013)
German	298,614	Universal Dependency Treebank v2.0 (UDT) (McDonald et al., 2013)
English	254,830	English Web Treebank v1.0 (EWT) (Silveira et al., 2014)
Italian	214,748	Italian Stanford Dependency Treebank (ISDT) (Simi et al., 2014)
<b>Finnish</b>	<b>202,085</b>	Turku Dependency Treebank (TDT) (Haverinen et al., 2013b)
Swedish	96,819	Talbanken (Nivre, 2014)
Hungarian	26,538	Szeged Treebank (Farkas et al., 2012)
Irish	23,686	Irish Dependency Treebank (IDT) (Lynn et al., 2014)

Table 4: Statistics of the UD Finnish treebank in comparison to the other treebanks included in the first UD data release.

TDT, but with `remnant` relations between *Maija* and *Mirja* and between *Pariisiin* and *Prahaan* in UD Finnish (see Figure 3). We applied a combination of custom scripts and manual reannotation to resolve empty nodes in the conversion of TDT to UD Finnish.

## 2.5 Annotation statistics

Table 4 shows token statistics for the 10 languages for which treebanks were included in the initial UD data release. With over 200,000 tokens, the UD Finnish treebank is in a mid-size cluster among the UD version 1 languages together with German, English and Italian. This is a relatively prominent position for Finnish, which until recently had no publicly available treebanks. We hope that the availability of this corpus will encourage further interest in Finnish dependency parsing.

## 3 Experiments

As discussed by de Marneffe et al. (2014) in the context of the Universal Stanford Dependencies which formed the basis on which UD was built, parsing accuracy has not been a major consideration in the definition of the scheme. In fact, a number of the design choices taken, such as the attachment of auxiliaries and prepositions as dependents rather than governors of their semantic head is known to result in a numerically worse parsing accuracy. Additionally, as the conversion is an automatic process, the resulting noise may have a detrimental effect on parsing accuracy as well. To quantify these effects, we carry out several parsing experiments, comparing the Stanford Dependencies annotation in TDT with its conver-

sion to the UD format. Further, since TDT now contains also fully manually annotated morphology, we will pay extra attention to morphological processing in the evaluation.

We base the experiments on the publicly available Finnish parsing pipeline.<sup>6</sup> The pipeline uses the CRF-based tagger Marmot (Müller et al., 2013), in conjunction with the two-level morphological analyzer OMorFi (Pirinen, 2008; Lindén et al., 2009). The morphological analyzer is used to provide the set of possible morphological readings (lemma, POS, and features) of every recognized word, which are subsequently given as features to the Marmot tagger. We initially apply a *hard* constraint approach, where the output of the tagger is used to select one of these readings (the reading with the highest overlap of tags and a priority for readings matching the main POS), effectively disambiguating OMorFi output. For words not recognized by OMorFi, the reading produced by Marmot is used as-is, and the wordform itself is used in place of the lemma. This has so far been the strategy taken when learning to parse Finnish (Bohnet et al., 2013). The tagged text is then parsed with the Mate tools graph-based dependency parser (Bohnet, 2010).<sup>7</sup>

As baseline, we consider the most recent Finnish dependency parser trained and evaluated on the original distribution of TDT. Note that the test sets differ: the baseline is evaluated on a test set matching the data it was trained on, which differs from the new test set in several aspects such as the treatment of named entities. The results are thus broadly comparable, but not directly so.

<sup>6</sup><http://turkunlp.github.io/Finnish-dep-parser/>

<sup>7</sup><https://code.google.com/p/mate-tools>

	POS	PM	FM	LAS	UAS
Baseline (Haverinen et al., 2013b)	94.3	90.5	89.0	81.4	85.2
Stanford Dependencies (SD)	96.3	93.4	90.3	80.1	84.1
Universal Dependencies (UD)	96.0	93.1	90.5	81.0	85.0
Pure Universal Dependencies (Pure UD)	96.0	93.1	90.5	81.5	84.7

Table 5: Results of the parsing experiments. *SD* refers to the morphological tagset and dependency relations as defined in TDT, *UD* to the universal tagset and relations, and *pure UD* to UD relations with no language-specific extensions. *POS* is the POS tagging accuracy, *PM* the accuracy of POS and all features, *FM* the accuracy of full morphology (including the lemma), and *LAS* and *UAS* are the standard labeled and unlabeled attachment score metrics.

	POS	PM	FM	LAS	UAS
Universal Dependencies (soft)	97.0	93.0	89.3	81.5	85.4
Universal Dependencies (hard-pos)	97.0	94.0	90.7	82.1	85.8
Pure Universal Dependencies (soft)	97.0	93.0	89.3	82.0	84.9
Pure Universal Dependencies (hard-pos)	97.0	94.0	90.7	82.7	85.4

Table 6: Results of the UD parsing experiments with the *soft* and *hard-pos* morphological tagging strategies.

The results are summarized in Table 5. Firstly, we see that all results are roughly comparable, meaning that the conversion to UD has had no major effect on the parsing accuracy. However, the attachment scores are somewhat lower compared to the baseline, likely due at least in part to the different treatment of named entities in the previously published baseline parser as opposed to both the newly introduced SD and UD versions of TDT. Unsurprisingly, the labeled attachment score is slightly higher for the pure UD scheme with no language-specific relations.

We additionally focused on morphological tagging. As TDT now contains manual morphological annotation, the analyses are no longer tightly bound to OMorFi as they were in the original release of TDT. We therefore consider also a *soft* constraint approach, where the tags given by Marmot are preserved, and OMorFi is only used to select the lemma (from the reading with the highest overlap of tags). This results in morphological analyses superior in POS accuracy but inferior in the prediction of full features. To address this issue, we implemented a new tagging strategy that applies the hard constraint only in cases where the predicted POS can be found among the analyses given by OMorFi (referred to as *hard-pos*). The results show an across-the-board improvement for this strategy as well as numerically the best scores for Finnish with the graph-based parser of Bohnet (2010) (Table 6).

## 4 Conclusions

We have presented Universal Dependencies (UD) for Finnish, detailing the application of general UD guidelines to the annotation of parts-of-speech, morphological features, and dependency relations in Finnish and introducing a conversion from the previously released Turku Dependency Treebank corpus into the UD Finnish treebank released in the first UD data release. We also performed experiments evaluating a state-of-the-art parser on both the source treebank, TDT, and the target UD Finnish treebank, finding that performance is slightly improved in the conversion, which supports both the accuracy of the conversion and the feasibility of UD as a parsing target.

All of the tools and resources described in this work are available under open licenses from <http://bionlp.utu.fi/ud-finnish.html>.

## Acknowledgments

This work was supported by the Kone Foundation and the Emil Aaltonen Foundation. Computational resources were provided by CSC - IT Center for Science. This paper builds on joint work with Jinho Choi, Marie-Catherine de Marneffe, Tim Dozat, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Joakim Nivre, Slav Petrov, Natalia Silveira, Reut Tsarfaty, and Dan Zeman.

## References

- Bejček, E., Panevová, J., Popelka, J., Straňák, P., Ševčíková, M., Štěpánek, J., and Žabokrtský, Z. (2012). Prague dependency treebank 2.5 – a revisited version of pdt 2.0. In *Proceedings of the 24th International Conference on Computational Linguistics (Coling 2012)*, pages 231–246.
- Bohnet, B. (2010). Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING'10*, pages 89–97.
- Bohnet, B., Nivre, J., Boguslavsky, I., Farkas, R., Ginter, F., and Hajič, J. (2013). Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics*, 1:415–428.
- Bosco, C., Montemagni, S., and Simi, M. (2013). Converting italian treebanks: Towards an italian stanford dependency treebank. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 61–69.
- de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. D. (2014). Universal Stanford Dependencies: A cross-linguistic typology. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, volume 14, pages 4585–4592.
- de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, volume 6, pages 449–454.
- Farkas, R., Vincze, V., and Schmid, H. (2012). Dependency parsing of hungarian: Baseline results and challenges. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 55–65.
- Hakulinen, A., Korhonen, R., Vilkuna, M., and Koivisto, V. (2004). *Iso suomen kielioppi*. Suomalaisen kirjallisuuden seura.
- Haverinen, K., Laippala, V., Kohonen, S., Missilä, A., Nyblom, J., Ojala, S., Viljanen, T., Salakoski, T., and Ginter, F. (2013a). Towards a dependency-based propbank of general finnish. In *Proceedings of the 19th Nordic Conference on Computational Linguistics (NoDaLiDa'13)*, pages 41–57.
- Haverinen, K., Nyblom, J., Viljanen, T., Laippala, V., Kohonen, S., Missilä, A., Ojala, S., Salakoski, T., and Ginter, F. (2013b). Building the essential resources for finnish: the Turku Dependency Treebank. *Language Resources and Evaluation*, pages 1–39.
- Kanerva, J., Luotolahti, J., Laippala, V., and Ginter, F. (2014). Syntactic n-gram collection from a large-scale corpus of internet finnish. In *Proceedings of the Sixth International Conference Baltic HLT*, pages 184–191.
- Lindén, K., Silfverberg, M., and Pirinen, T. (2009). HFST tools for morphology — an efficient open-source package for construction of morphological analyzers. In *State of the Art in Computational Morphology*, volume 41 of *Communications in Computer and Information Science*, pages 28–47.
- Lynn, T., Foster, J., Dras, M., and Tounsi, L. (2014). Cross-lingual transfer parsing for low-resourced languages: An Irish case study. In *Proceedings of the First Celtic Language Technology Workshop*, pages 41–49.
- McDonald, R., Nivre, J., Quirnbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Bertomeu Castelló, N., and Lee, J. (2013). Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 92–97.
- Müller, T., Schmid, H., and Schütze, H. (2013). Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.
- Nivre, J. (2014). Universal Dependencies for Swedish. In *SLTC 2014*.
- Nivre, J., Bosco, C., Choi, J., de Marneffe, M.-C., Dozat, T., Farkas, R., Foster, J., Ginter, F., Goldberg, Y., Hajič, J., Kanerva, J., Laippala, V., Lenci, A., Lynn, T., Manning, C., McDonald, R., Missilä, A., Montemagni, S., Petrov, S., Pyysalo, S., Silveira, N., Simi, M., Smith, A., Tsarfaty, R., Vincze, V., and Zeman, D. (2015). Universal dependencies 1.0.
- Nivre, J., Choi, J., de Marneffe, M.-C., Dozat, T.,

- Ginter, F., Goldberg, Y., Hajič, J., Manning, C., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2014). Universal dependencies documentation 1.0.
- Petrov, S., Das, D., and McDonald, R. (2012). A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, pages 2089–2096.
- Pirinen, T. (2008). Suomen kielen äärellistilainen automaattinen morfologinen jäsennin avoimen lähdekoodin resurssien. Master's thesis, University of Helsinki.
- Silveira, N., Dozat, T., de Marneffe, M.-C., Bowman, S., Connor, M., Bauer, J., and Manning, C. D. (2014). A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.
- Simi, M., Bosco, C., and Montemagni, S. (2014). Less is more? towards a reduced inventory of categories for training a parser for the italian stanford dependencies. In *Proceedings of LREC 2014*.
- Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.
- Sulkala, H. and Karjalainen, M. (1992). *Finnish. Descriptive Grammar Series*. Routledge, London.
- Tsarfaty, R. (2013). A unified morpho-syntactic scheme of stanford dependencies. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 578–584.
- Voutilainen, A. (2011). FinnTreeBank: Creating a research resource and service for language researchers with Constraint Grammar. In *Proceedings of the NODALIDA 2011 workshop Constraint Grammar Applications*.
- Zeman, D. (2008). Reusable tagset conversion using tagset drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, pages 213–218.





**Jenna Kanerva & Filip Ginter & Tapio Salakoski**  
**Universal Lemmatizer: A Sequence to Sequence Model for**  
**Lemmatizing Universal Dependencies Treebanks**

Natural Language Engineering. 2021; 27(5):545–574.



ARTICLE

# Universal Lemmatizer: A sequence-to-sequence model for lemmatizing Universal Dependencies treebanks

Jenna Kanerva<sup>\*</sup> , Filip Ginter and Tapio Salakoski

TurkuNLP Group, Department of Future Technologies, University of Turku, Turku, Finland

<sup>\*</sup>Corresponding author. Email: [jmnybl@utu.fi](mailto:jmnybl@utu.fi)

(Received 1 February 2019; revised 2 March 2020; accepted 2 March 2020; first published online 27 May 2020)

## Abstract

In this paper, we present a novel lemmatization method based on a sequence-to-sequence neural network architecture and morphosyntactic context representation. In the proposed method, our context-sensitive lemmatizer generates the lemma one character at a time based on the surface form characters and its morphosyntactic features obtained from a morphological tagger. We argue that a sliding window context representation suffers from sparseness, while in majority of cases the morphosyntactic features of a word bring enough information to resolve lemma ambiguities while keeping the context representation dense and more practical for machine learning systems. Additionally, we study two different data augmentation methods utilizing autoencoder training and morphological transducers especially beneficial for low-resource languages. We evaluate our lemmatizer on 52 different languages and 76 different treebanks, showing that our system outperforms all latest baseline systems. Compared to the best overall baseline, UDPipe Future, our system outperforms it on 62 out of 76 treebanks reducing errors on average by 19% relative. The lemmatizer together with all trained models is made available as a part of the Turku-neural-parsing-pipeline under the Apache 2.0 license.

**Keywords:** Lemmatization; Universal Dependencies; Parsing; Sequence-to-sequence model

## 1. Introduction

Lemmatization is a process of determining a base or dictionary form (lemma) for a given surface form. Traditionally, word base forms have been used as input features for various machine learning tasks such as parsing, but also find applications in text indexing, lexicographical work, keyword extraction, and numerous other language technology-enabled applications. Lemmatization is especially important for languages with rich morphology, where a strong normalization is required in applications. Main difficulties in lemmatization arise from encountering previously unseen words during inference time as well as disambiguating ambiguous surface forms which can be inflected variants of several different base forms depending on the context.

The classical approaches to lemmatizing highly inflective languages are based on two-level morphology implemented using finite state transducers (FSTs) (Koskenniemi 1984; Karttunen and Beesley 1992). FSTs are models encoding vocabulary and string rewrite rules for analyzing an inflected word into its lemma and morphological tags. Due to surface form ambiguity, the FST encodes all possible analyses for a word, and the early work on context-sensitive lemmatization was based on disambiguating the possible analyses in the given context (Smith, Smith, and Tromble 2005; Aker, Petrak, and Sabbah 2017; Liu and Hulden 2017).

The requirement of having a predefined vocabulary is impractical especially when working with Internet or social media texts where the language variation is high and adaptation fast.

Therefore, there has been an increasing interest in the application of context-sensitive machine learning methods that are able to deal with open vocabulary.

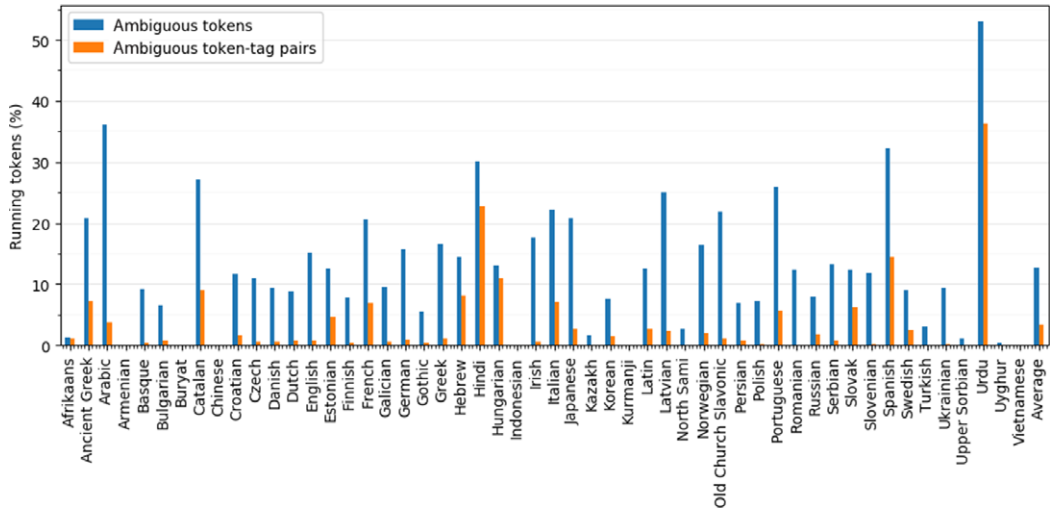
In this paper, we present a sequence-to-sequence lemmatizer with a novel context representation. This method was used as part of the TurkuNLP submission (Kanerva *et al.* 2018) in the CoNLL-18 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman *et al.* 2018), where it ranked 1st out of 26 participants on the lemmatization subtask. In addition to plain lemmatization, the system ranked 1st on the bi-lexical dependency score evaluation metric as well, a metric combining evaluation of both lemmatization and syntactic dependencies. Our Shared Task work is extended in several directions. First, we analyze and justify the particular context representation used by the system using data from 52 languages; second, we carry out comparison to state-of-the-art lemmatization methods; third, we test and evaluate two different data augmentation methods for automatically expanding training data sizes; and finally, we release the system together with models for all 52 languages as a freely available parsing pipeline, containerized using Docker for ease of use.

The rest of the paper is structured as follows. In Section 2, we discuss the surface form ambiguity problem in the context of lemmatization, as well as present a data-driven study for justifying our contextual representation for resolving the problem. In Section 3, we describe the most important related work. In Section 4, we present our problem setting, model architecture, and implementation. Experimental setups for our main evaluation as well as results are given in Sections 5 and 6. In Section 7, we describe our data augmentation studies to increase training set sizes leading to a higher prediction accuracy. In Section 8, we summarize the results as well as discuss the practical issues related to our method, most importantly prediction speed and software release. Finally, we conclude the paper in Section 9.

## 2. Lemmatization ambiguity and morphosyntactic context

Lemmatization methods can roughly be divided into two categories, context-aware methods where the lemmatization system is aware of the sentence context where the word appears, and methods where the system is lemmatizing individual words without contextual information. The advantage in the former approach is the ability to correctly lemmatize ambiguous words based on the contextual information while the latter is only able to either give one lemma for each word even though its lemmatization can vary in different contexts, or list all alternatives. While some of the ambiguous words, such as *love* in the verb-noun contrast (*I love you* vs. *Love is all you need*), are assigned the same lemma (*love* in this case rather than *to love*), it is not always the case. For example, the English word *lives* receives a different lemma depending on the part of speech (*live* vs. *life*). Additionally, words can be ambiguous within a single part-of-speech class. For example, in Finnish the word *koirasta* is always a noun but depending on the grammatical case it should be lemmatized to *koira* (*a dog* inflected in elative case) or to *koiras* (*a male* inflected in partitive case). Note that the knowledge of the part-of-speech and inflectional tags, that is, morphosyntactic features of the word, is sufficient to correctly lemmatize these two abovementioned examples. This holds for the majority of cases, with rare exceptions. For example, the Finnish word *paikkoja* is a noun in plural partitive, but it can be an inflection of two different lemmas, *paikka* (*a place* or *patch*) or *paikko* (*a spare* in bowling). In these rare cases, the meaning, and therefore the correct lemma, can only be derived from the semantic context, that is, the actual meaning or topic of the sentence.

Bergmanis and Goldwater (2018) did a careful evaluation of lemmatization model effectiveness with and without contextual information. They show that including a sliding window of nearby characters significantly improves the performance compared to the context-free version of the same system. However, they only evaluate the system using a textual context (i.e. *n* characters/words before and after the word to be lemmatized). Suspecting that this lexical context

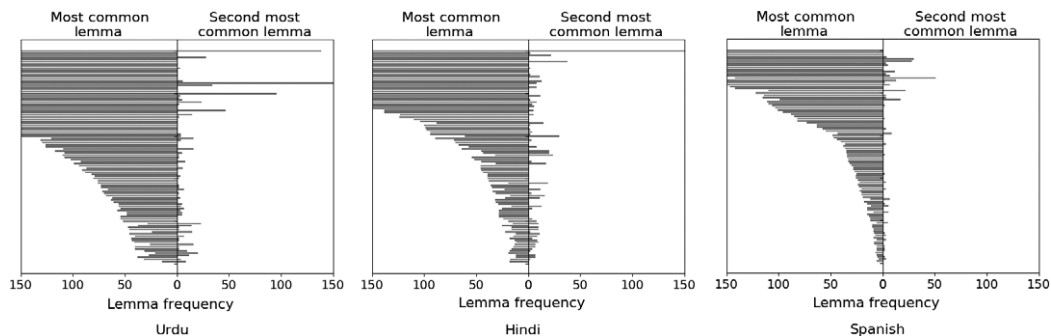


**Figure 1.** Percentage of running tokens with ambiguous lemma and token-tag pairs with ambiguous lemma calculated from the UD v2.2 training data. An ambiguous token is a word occurring with more than one lemma in the training data, whereas an ambiguous token-tag pair is a (word, morphosyntactic tags) -tuple occurring with more than one lemma in the training data. All treebanks of one language are pooled together.

representation suffers from sparseness, we hypothesize that the morphosyntactic features will uniquely disambiguate the lemma in all but the rarest of cases, and can serve as a more practical, dense context representation for the lemmatization task. In order to establish how uniquely the features disambiguate the lemma, we measure different levels of ambiguity on the Universal Dependencies (UD) v2.2 treebanks and present the results in Figure 1. We measure how many times a (word, morphosyntactic tags) -tuple is seen with more than one lemma compared to how many times a plain word is seen with more than one lemma in the training data.

We can see that the proportion of ambiguous lemmas drastically drops for most languages when morphosyntactic tags are taken into account, on average the token-tag pair ambiguity being close to 3% of running tokens, while plain token ambiguity is close to 12%. For more than half of the languages, the ambiguity drops below 1% of running tokens, to the level which does not pose an issue anymore, or, from a different point of view, can be expected to cause an issue to any machine learning system due to the rareness of the words involved as we will demonstrate shortly. However, for few languages the ambiguity remains on surprisingly high level, especially for Urdu (36%) and Hindi (22%), both being Indo-Aryan languages and closely related to each other, as well as for Spanish (14%), a Romance language. To shed some light specifically on these three languages, we plot in Figure 2 the frequencies of most common and second most common lemmas for the 100 most common ambiguous words. For all three languages, and all but a handful of words, the distribution is extremely imbalanced with only a small number of occurrences of the less frequent lemma. When investigating similar cases in languages we are familiar with, we can see that in addition to real ambiguities in many cases these turn out to be annotation inconsistencies. For example, while the word *vs.* as adposition has only one meaning in the English training data and therefore should also have only one lemma, it is lemmatized 17 times as *vs.* and once as *versus*. Similarly, most of the ambiguous cases in the Finnish data are inconsistencies in the placement of compound boundary markers. Even with the real ambiguities, it is debatable whether heavily skewed distributions, where the most common lemma can be several orders of magnitude more common, can be learned given the minimal number of training examples for the rarer lemma.

In the light of these findings, we therefore argue that the part-of-speech and rich morphosyntactic features are, from the practical standpoint of building a multilingual lemmatization system,



**Figure 2.** Frequency comparison of the most common and the second most common lemmas in the training data for words which are ambiguous at the word-tag level. The top-100 most common ambiguous words are shown for Urdu (left), Hindi (middle), and Spanish (right), the three languages with the highest ambiguity rate in Figure 1.

sufficient to resolve the vast majority of ambiguous lemmatizations in the vast majority of the 52 languages covered by the UD data set.

### 3. Related work

The most common machine learning approaches to lemmatization are based on edit-tree classification, where all possible edit trees or word-to-lemma transformation rules are first gathered from the training data, and then a classifier is trained to choose the correct one for a given input word. These methods do not require that the input word is known in advance as long as the correct edit pattern is seen during training. Edit-tree classifiers are used, for example, in Müller *et al.* (2015), Straka, Hajic, and Straková (2016), and Chakrabarty, Pandit, and Garain (2017), and the sentence-context for resolving ambiguous words can be incorporated into these classifiers, for example, by using global sentence features (Müller *et al.* 2015) or contextualized token representations (Straka *et al.* 2016; Chakrabarty *et al.* 2017; Straka 2018b).

Many recent works build on the sequence-to-sequence learning paradigm. Bergmanis and Goldwater (2018) present the Lematus context-sensitive lemmatization system, where the model is trained to generate the lemma from a given input word one character at a time. Additionally, a context of 20 characters in each direction is concatenated with the input word, resulting in a 12% relative error decrease compared to only the word being present in the input. The Lematus system outperforms other context-aware lemmatization systems, including Chrupała, Dinu, and Van Genabith (2008), Müller *et al.* (2015), and Chakrabarty *et al.* (2017), and can be seen at the time of writing as the current state of the art on the task. However, the task is naturally an active research area with new directions pursued, for example, by Kondratyuk *et al.* (2018).

The 2018 CoNLL Shared Task on multilingual parsing included lemmatization as one of the objectives, and has given rise to a number of machine learning approaches. Together with our work and the abovementioned edit-tree classifier of Straka (2018b), the Stanford system (Qi *et al.* 2018) ranked among the top three performing systems on large treebanks in the Shared Task. In the Stanford system, words whose lemma cannot be looked up in a dictionary are lemmatized using a sequence-to-sequence model without any additional context information.

Sequence-to-sequence models have also been widely applied in the context of morphological reinfection, the reverse of the lemmatization task. In the CoNLL-SIGMORPHON 2017 Shared Task on Universal Morphological Reinfection (Cotterell *et al.* 2017), the objective was to generate the inflected word given a lemma and morphosyntactic tags. Here several of the top-ranking systems were based on sequence-to-sequence learning (Kann and Schütze 2017a; Bergmanis *et al.* 2017). The entry of Östling and Bjerva (2017) additionally tried to boost the inflection generation by learning the primary morphological reinfection objective jointly with the reverse task of lemmatization and tagging.

## 4. Methods

Taking inspiration from the top systems in the CoNLL-SIGMORPHON 2017 Shared Task, we cast lemmatization as a sequence-to-sequence rewrite problem where lemma characters are generated one at a time from the given sequence of word characters and morphosyntactic tags. We diverge from previous work on lemmatization by utilizing morphosyntactic features predicted by a tagger to represent the salient information from the context, instead of using, for example, contextualized word representations or sliding window of text. We modify the usual order of a parsing pipeline to include the lemmatizer as the last step of the pipeline, running after the tagger and thus making it possible to access the predicted part-of-speech and morphological features at the time of lemmatization. In this study, we use the part-of-speech tagger of Dozat, Qi, and Manning (2017) modified to predict also morphological features (Kanerva *et al.* 2018). More detailed discussion of the tagger is included in Section 5.1.2.

The input of our sequence-to-sequence lemmatizer model is the sequence of characters of the word together with the sequence of its morphosyntactic tags, while the output is the sequence of lemma characters. In the UD representation, three different columns are available for morphosyntactic tags: universal part-of-speech (UPOS), language-specific part-of-speech (XPOS), and morphological features, a sorted list of feature category and value pairs (FEATS). All three are used in the input together with the word characters. For example, the input and output sequences for the English word *lives* as a noun are the following:

```
INPUT: l i v e s UPOS=NOUN XPOS=NNS Number=Plur
OUTPUT: l i f e
```

Once cast in this manner, essentially any of the recent popular sequence-to-sequence model architectures can be applied to the problem. Similarly to the Lematus system, we rely on an existing neural machine translation model implementation, in our case OpenNMT: Open-Source Toolkit for Neural Machine Translation (Klein *et al.* 2017).

### 4.1 Sequence-to-sequence model

The model implemented by OpenNMT is a deep attentional encoder–decoder network. The encoder uses learned character and tag embeddings, and two bidirectional long short-term memory (LSTM) layers to encode the sequence of input characters and morphosyntactic tags into a same-length sequence of encoding vectors. The sequence of output characters is generated by a decoder with two unidirectional LSTM layers with input feeding attention (Luong, Pham, and Manning 2015b) on top of the encoder output. The full model architecture is illustrated in Figure 3.

An important requirement for sequence-to-sequence models is the ability to correctly deal with out-of-vocabulary (OOV) items at inference time. For example, in machine translation foreign person and place names should often be copied into the output sequence, which is not possible if the generation is based on a straightforward classification over output vocabulary learned during training. In the case of lemmatization, this issue manifests itself as characters not seen during training. Since in some languages foreign names inflect, copying full words that contain OOV characters is not a sufficient solution. For instance, a Finnish lemmatizer model trained on a typical Finnish corpus will have a vocabulary of mostly Scandinavian characters, and will be unable to correctly lemmatize the case-inflected Czech name *Růžičkalla* into *Růžička*.

In machine translation, the problem of OOV words is for the most part solved using byte pair encoding or other subword representations, reducing vocabulary size and handling inference-time unknown words (as unknown words can be split into known subwords) (Sennrich, Haddow, and Birch 2016). As the lemmatizer operates on the level of characters, indivisible into smaller units, we instead rely on an alternative technique whereby the model is trained to predict an unknown

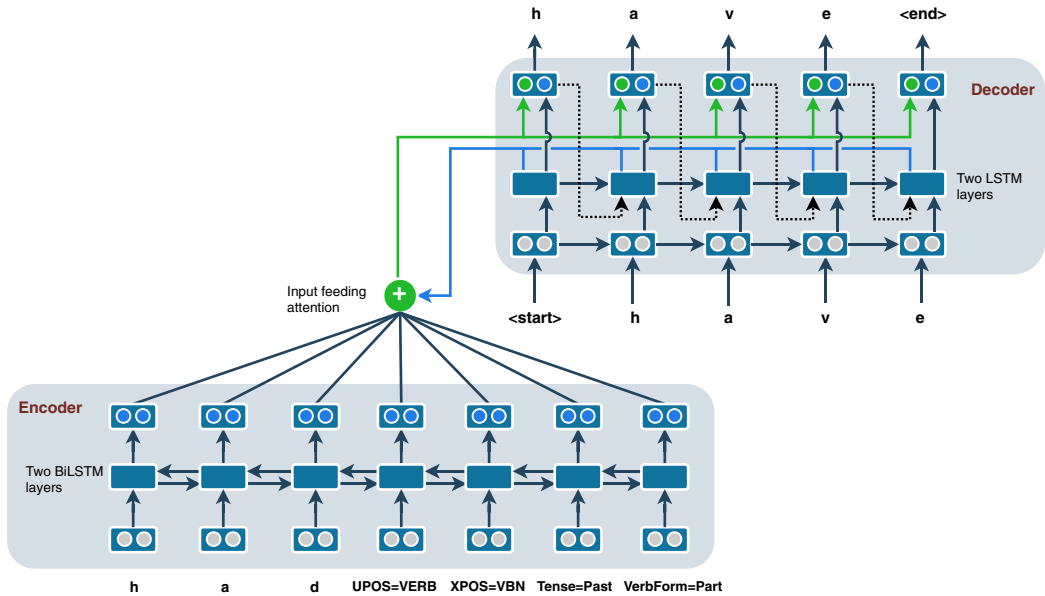


Figure 3. Our encoder–decoder model architecture.

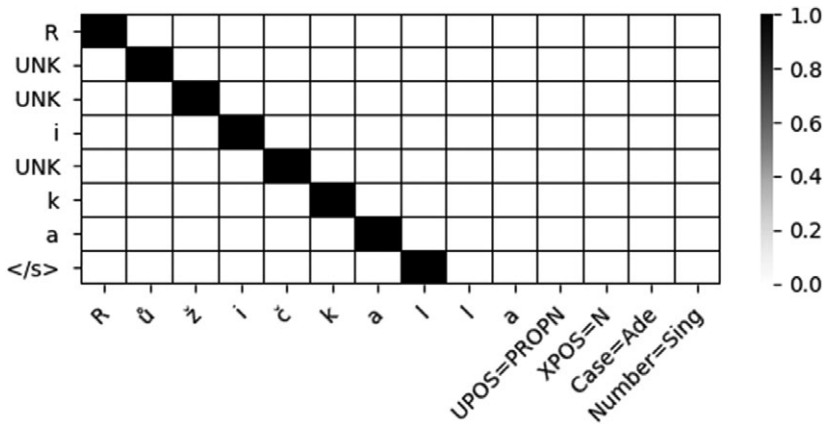


Figure 4. Visualization of the step-wise attention weights (actual system output), where the x-axis corresponds to the input sequence and the y-axis to the generated output sequence. In post-processing, each generated UNK symbol is replaced with the input symbol that has the maximal attention at the respective time step.

symbol UNK for rare and unseen characters, and as a post-processing step, each such UNK symbol is subsequently substituted with the input symbol with the maximal attention value of the model at that point (Luong et al. 2015a; Jean et al. 2015). For instance, for the inflected name *Růžičkalla*, we would get

INPUT: R ù ž i č k a l l a UPOS=PROPN XPOS=N Case=Ade Number=Sing  
 OUTPUT: R UNK UNK i UNK k a

as the initial output of the system, later post-processed to the correct lemma *Růžička* based on attention weights visualized in Figure 4.

## 5. Evaluation

Next we carry out an extensive evaluation of the lemmatization framework on 52 different languages with varying lemmatization complexity and training data sizes. We compare our system to several competitive lemmatization baselines. First, we give a detailed description of our experimental setup, the baseline systems, and model parameters, and after that, we present the evaluation results.

### 5.1 Data and tools

#### 5.1.1 UD treebanks

We base our experiments on UD v2.2 (Nivre *et al.* 2018), a multilingual collection of 122 morpho-syntactically annotated treebanks for 71 languages, with cross-linguistically consistent annotation guidelines, including also gold standard lemma annotation (Nivre *et al.* 2016). The UD treebanks therefore allow us to test the lemmatization methods across diverse language typologies and training data sizes, ranging from a little over 100 to well over 1 million tokens. We restrict the data to the subset of 82 treebanks (57 languages) used in the CoNLL-18 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman *et al.* 2018). In addition to allowing a direct comparison with the state-of-the-art parsing pipelines participating in the Shared Task, the treebanks from this subset all have a test set of at least 10,000 tokens, ensuring a reliable evaluation. Note that even though the test set is always at least 10,000 tokens, training sets may be considerably smaller, in several instances about 100 tokens.

Furthermore, it was also necessary to remove two treebanks with no lemma annotation (Old French-SRCMF and Thai-PUD) and four treebanks with no training data (Breton-KEB, Faroese-OFT, Japanese-Modern and Naija-NSC). The four parallel “PUD” treebanks included in the Shared Task (Czech-PUD, English-PUD, Finnish-PUD, and Swedish-PUD, each including the same 1000 sentences translated into the target language and annotated into UD) do not have dedicated training data, but can be used as additional test sets for models trained on the Czech-PDT, English-EWT, Finnish-TDT, and Swedish-Talbanken treebanks, which are sufficiently similar in annotation style. Altogether, we therefore evaluate on 76 treebanks representing 52 different languages. During evaluation, we show results separately for several different groups categorizing treebanks by size or other properties. These groups are *PUD* for 4 additional parallel test sets, *big* for 60 treebanks with more than 10,000 tokens of training and 5000 tokens of development data, *small* for 7 treebanks with reasonably sized training data but no additional development data, and *low resource* for 5 treebanks with only a tiny sample of training data (around 20 sentences) and no development data. These are the same treebank groups as defined in CoNLL-18 Shared Task.

To ensure that treebanks in the *small* and *low-resource* categories also have a development set for hyperparameter tuning and model selection, we adopt the data split provided by the Shared Task organizers, which creates the development set from a portion of the training data when necessary (Straka 2018a). This data split was also used to train the Shared Task baseline model, one of the systems we compare our results to. The final numbers are always reported on the held-out test set directly specified in the UD release for each treebank. The original test section of the UD data is never used in system training and development, as suggested by the data providers and so as to be able to distribute the trained models for further comparison. For this reason, we also decided not to apply N-fold cross-validation for low-resource treebanks, which otherwise would have been an option to decrease variance in the results. Furthermore, the training and development set split is also kept fixed as the development data are used only for early stopping and model selection, which we do not expect to greatly affect the numbers, and hyperparameters are not tuned separately for each treebank.

#### 5.1.2 Part-of-speech and morphological tagger

As the input of our lemmatizer is a word together with its part-of-speech and morphosyntactic features, we need a tagger to predict the required tags before the word can be lemmatized. We use



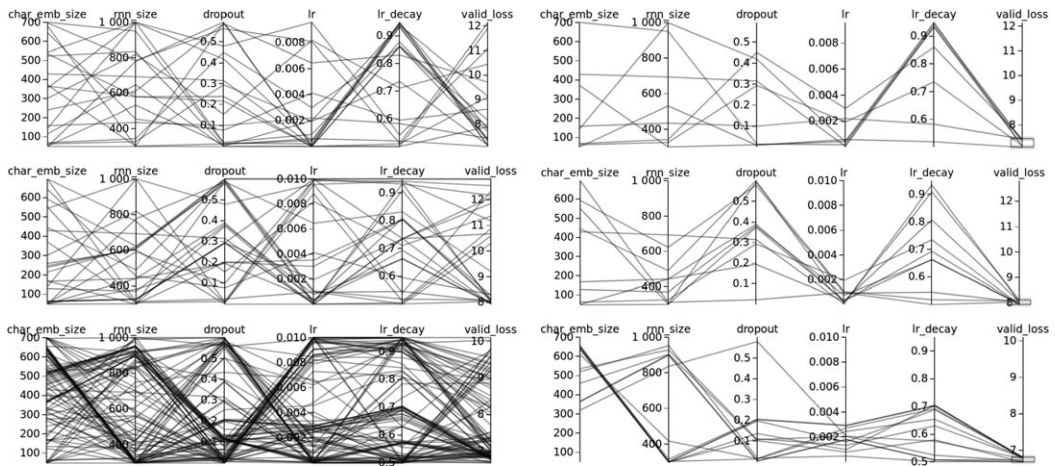
the one by Kanerva *et al.* (2018), which is based on the winning Stanford part-of-speech tagger (Dozat and Manning 2017; Dozat *et al.* 2017) from the CoNLL-17 Shared Task on multilingual parsing (Zemen *et al.* 2017). The tagger has two classification layers (predicting UPOS and XPOS) over tokens in a sentence, where tokens are first embedded using a sum of learned, pretrained and character-based LSTM embeddings, which are then encoded with a bidirectional LSTM to create a sequence of contextualized token representations. The classification layers are trained jointly on top of these shared token representations. By default, the original tagger does not predict the rich morphosyntactic features (FEATS column in CoNLL-U format). To this end, in Kanerva *et al.* (2018) we modified the tagger training data by concatenating the morphosyntactic features with the language-specific part-of-speech tag (XPOS), thereby forcing the tagger to predict the XPOS tag and all morphosyntactic features as one multi-class classification problem. For example, in Finnish-TDT the original XPOS value N and FEATS value Case=Nom|Number=Sing are concatenated into one long string XPOS=N|Case=Nom|Number=Sing which is then predicted by the tagger. The morphological features are sorted so as to avoid duplicating label strings having the same tags in different order. After prediction, the morphosyntactic features are extracted into a separate column. The evaluation in Kanerva *et al.* (2018) shows that this data manipulation technique does not harm the prediction of the original XPOS tag, and accuracy of morphosyntactic feature prediction (FEATS field) is comparable to the state of the art in the CoNLL-18 Shared Task, ranking 2nd in the evaluation metric combining both morphosyntactic features and syntactic dependencies, and 3rd in the evaluation of plain morphosyntactic features. In our preliminary experiments, we expected the complex morphology of some languages to result in a large number of very rare feature strings if combined in such a simple manner. We tested several models, for instance, predicting a value for each category separately (e.g. *Nominative* for *Case*) from a shared representation layer. However, the results were surpassed by the simple concatenation of morphological features. The conclusion of this experiment was that even though some languages have many unique feature combinations (number of unique combinations ranging from 15 to 2500) the most common ones cover the vast majority of the data, with the rare classes having no practical effect on the prediction accuracy (more detailed discussion is given in Kanerva *et al.* 2018).

## 5.2 Parameter optimization

To optimize the hyperparameters of our lemmatization models, we use the RBFOpt library designed for optimizing complex black-box functions with costly evaluation (Costa and Nannicini 2018). Different values of embedding size, recurrent layer size, dropout, learning rate, and learning rate decay parameters are experimented with. We let the RBFOpt optimizer run for 24 hours on 3 different treebanks, completing about 30 training runs for Finnish and English, and about 300 for the much smaller Irish treebank. The findings are visualized in Figure 5: On the left side of the figure, all different runs completed by the optimizer are shown as a parallel coordinates graph, while on the right side we use a validation loss filter to show only those runs that result in low validation loss values. From this, we can more easily determine the optimal parameter ranges and their mutual relationship.

Based on these optimizer runs, the lemmatization models seem to be moderately stable, most of the parameter values having individually only a small influence on the resulting validation loss, once the RBFOpt optimizer finds the appropriate region in the parameter space. The learning rate parameter (1r column) appears to have the largest impact, where lower learning rate values generally work better. Overall, the learning is stable across the parameter space, and the parameter optimization does not play a substantial role. Even default values as defined in the OpenNMT toolkit worked comparatively well.

In the final experiments, apart from the batch size, uniform hyperparameter settings based on the observations of the three optimization runs are used for all treebanks. We set the embedding size to 500, dropout to 0.3, recurrent size to 500, and we use the Adam optimizer (Kingma and Ba



**Figure 5.** Parallel coordinates graphs for visualizing hyperparameter optimizer runs for three different treebanks (top: English, middle: Finnish, bottom: Irish). On the left side of the figure are all optimizer runs completed during the 24-hour time window, while on the right side these runs are filtered based on the validation loss to demonstrate parameter ranges resulting in low validation loss values.

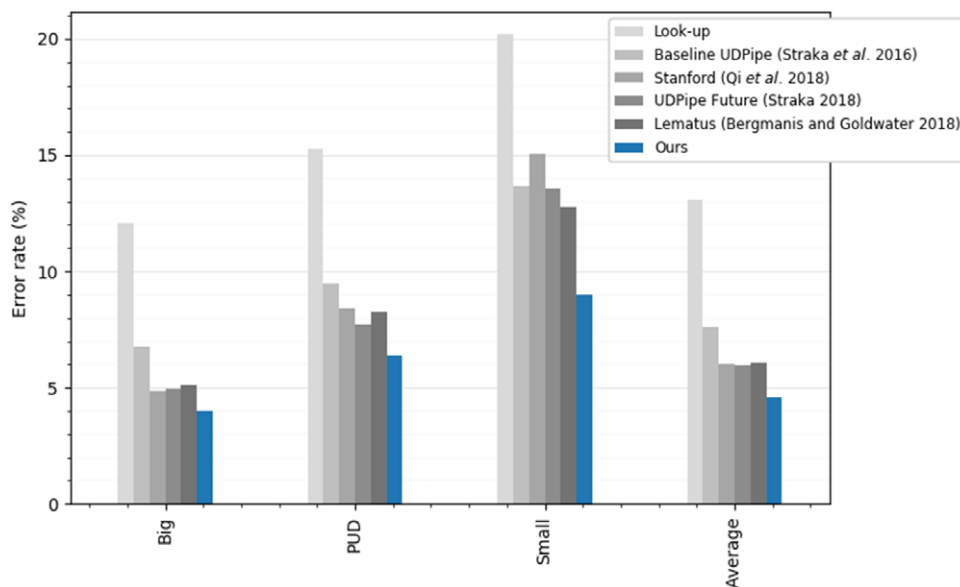
2015) with initial learning rate of 0.0005 and learning rate decay with 0.9 starting after 20 epochs. All models are trained for 50 epochs, but for smaller treebanks we decrease the minibatch size to increase the number of updates applied during training. Our default minibatch size is 64, but for treebanks with less than 2000 training sentences and less than 200 training sentences, we use 32 and 6, respectively. Models usually converge around epochs 30–40, and final models are chosen based on prediction accuracy on the validation loss set. During prediction time we use beam search with beam size 5.

### 5.3 Baselines

We compare our lemmatization performance to several, recent baseline systems. *Baseline UDPipe* (Straka *et al.* 2016) is the organizers’ baseline parsing pipeline from the CoNLL-18 Shared Task, which, due to its easy usability and availability of pretrained models, has been the go-to tool for parsing UD data. *UDPipe Future* (Straka 2018b) is an updated version of the baseline UDPipe pipeline ranking high across the CoNLL-18 ST evaluation metrics. Both UDPipe versions have a lemmatizer based on the edit-tree classification method. The *Stanford* system (Qi *et al.* 2018) is a dictionary look-up followed by a context-free sequence-to-sequence lemmatizer for words unseen in the training data. Together with our entry, *UDPipe Future* and *Stanford* form the top three performing entries in the lemmatization evaluation of the CoNLL-18 ST on the big treebank category. In addition to top ranking systems from the CoNLL-18 ST, we also compare to the context-aware *Lematus* sequence-to-sequence lemmatizer (Bergmanis and Goldwater 2018) which outperformed all its baselines in the earlier studies, and can be seen as a current state of the art in lemmatization research. Our final baseline (*Look-up*) is a simple look-up table, where lemmas are assigned based on the most common lemma seen in the training data, while unknown words are simply copied unchanged to the lemma field.

Results for the baseline systems from the CoNLL-18 ST (*Baseline UDPipe*, *UDPipe Future* and *Stanford*) are obtained directly from the official ST evaluation,<sup>a</sup> while the *Lematus* models are reimplemented using the OpenNMT toolkit to overcome the experimental differences between

<sup>a</sup> Evaluation results are available at <http://universaldependencies.org/conll18/results-lemmas.html>.



**Figure 6.** Test set word-level error rates for our system as well as all baseline systems divided into three different treebank groups, big, PUD, and small, as well as macro-average over all treebanks belonging to these groups.

this and the original study and performance issues regarding the original implementation.<sup>b</sup> To mimic the CoNLL-18 ST lemmatization evaluation settings, where lemmas are evaluated on top of the predicted sentence and word segmentation, we apply the segmentation of the *Baseline UDPipe* system (Straka 2018a) for our lemmatizer as well as for the *Lematus* and *Look-up* baselines. The *UDPipe Future* and *Stanford* systems instead have their own built-in segmenters. However, Straka (2018b) reports that when using the same segmentation as in our pipeline, the lemmatization accuracy of *UDPipe Future* decreased by 0.03pp overall, showing that the difference between our and *UDPipe Future* segmentation is not significant. For the *Stanford* system, comparable numbers are not available, and we need to rely on the official Shared Task evaluation.<sup>c</sup>

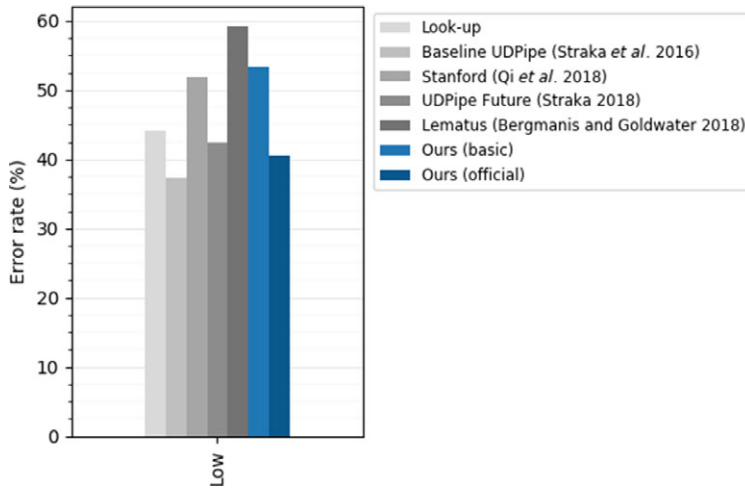
## 6. Results

The results are shown in Figure 6, where we measure word-level error rates separately on three treebank categories, *big*, *PUD*, and *small*, as well as macro-average error rate over all treebanks belonging to these three categories.

On all three categories, our system outperforms all the baselines with an overall error rate of 4.61 (macro-average across the treebanks in the three categories). Compared to the second best overall system, *UDPipe Future*, our error rate is 1.35 absolute percent point lower, reducing errors by 23% within these three treebank categories. The widest margin from our system to the second best systems is in the *small* treebank category where our system reduces errors by 30%, from 12.75 to 8.98, compared to the second best *Lematus* system. The simplistic *Look-up* baseline is clearly worse than all other systems, reflecting that plain memorizing training tokens and fallback copying unknowns is not a sufficient strategy for language universal lemmatizer. The three most recent

<sup>b</sup> The original implementation relies on the outdated Theano backend which is no longer compatible with our GPU servers.

<sup>c</sup> Note that the Stanford system official results are affected by a known segmentation bug. Overall lemmatization results reported by the Stanford team for their corrected system improve its performance from  $-2.92$ pp to  $-2.07$ pp difference to our system, that is, not affecting the overall conclusions.



**Figure 7.** Test set macro-average error rates of five low-resource category treebanks for two our models as well as all baseline systems.

baseline systems (*Stanford*, *UDPipe Future* and *Lematus*) perform evenly in terms of average error rate, outperforming the older *Baseline UDPipe*.

The fourth treebank category used in the CoNLL-18 ST is *low-resource*, where only a tiny training data sample is available, usually around 20 sentences. Results for this group are given separately in Figure 7, where we measure macro-average word-level error rate over the five treebanks belonging to this category. Few dozens of training sentences cannot be expected to result in a well-performing lemmatization system, and indeed, all systems have error rates near 40%–50%, where almost half of the tokens are lemmatized incorrectly. Here even the *Look-up* baseline performs comparably to the other systems, which is for the most part caused by the fallback copying of the unknown words unchanged to the lemma field, and therefore getting the easy words correct. For our system, we report two different runs, *basic* is trained purely on the tiny training data sample, while *official* is our official submission for the CoNLL-18 ST, where we experimented with preliminary data augmentation methods for automatically enriching the tiny training data sample with words analyzed by morphological transducers. The two lowest average error rates in the low-resource category are achieved by the two different versions of UDPipe (*UDPipe Baseline* and *UDPipe Future*), both belonging to the category of edit-tree classification systems. Systems based on sequence-to-sequence learning (*Stanford*, *Lematus*, and *ours*) are hypothesized to be more data hungry, and these systems indeed achieve clearly worse results in the low-resource category, all making more errors than correct predictions. However, when we include the additional training data obtained with data augmentation methods, we are able to boost our performance (*Our official*) to the level of the two edit-tree classification systems reducing errors by 24% compared to our basic models. Nevertheless, as all results are about the same level as the simple *Look-up* baseline, the achieved improvement is mostly theoretical.

## 7. Training data augmentation

In our initial attempt to improve lemmatization performance on the low-resource languages in the CoNLL-18 Shared Task, we observed a substantial improvement over our basic run when the morphological transducers are used to generate additional training data. However, the overall accuracy of those data sets is below the limits of usable real-world systems and thus the seen improvements are more theoretical than practical. Next, we investigate whether automatic training data augmentation methods are useful for languages with much better baseline accuracy to improve

lemmatization performance in a real-life setting as well. We test two different methods on a full set of treebanks suitable for a given method. First, we apply an autoencoder style secondary learning objective, where the lemmatizer model is trained to repeat the given input sequence without any modification. The benefit of such objective is to support the stem generation without requiring any additional resources. Second, we repeat the experiment with the morphological transducers for all languages which have an Apertium morphological transducer available. We generate additional inflection–lemma pairs based on the known vocabulary and inflection paradigms encoded as a transducer, and these new training examples are then mixed with the original training data. Next, we explain both data augmentation methods in detail, and afterwards compare the results.

### 7.1 Autoencoding random strings

In our first data augmentation method, we apply joint learning of autoencoding and lemmatization. The basis of the required work in sequence-to-sequence lemmatization is the ability to repeat the word stem in the output generation. As suggested by Kann and Schütze (2017b) in the context of morphological reinflection, we hypothesize that learning to repeat the input characters as a secondary task with additional training examples could simplify the lemmatization complexity the model has to learn especially for treebanks with less training data. If the model is taught separately to repeat the input characters in the generated output, the actual lemmatization rewriting task could be learnable with less training material. In particular, this approach should be able to help in low-resource settings when the amount of training data is not necessarily sufficient for learning the complex task from scratch.

Following the autoencoding idea of Kann and Schütze (2017b), we enrich our lemmatization training data for each treebank by adding randomly generated strings where the input and output sequences are verbatim copies. These random strings are not equipped with any morphosyntactic tags, but instead a special tag is added to give the model the ability to distinguish these from the actual lemmatization examples to avoid confusion. Each random string is generated by sampling with replacement 3–12 characters individually from the known character vocabulary with character probabilities calculated from the training data, producing word-like items of varying lengths. However, we force each character in the vocabulary to be sampled at least once to better cover the known character vocabulary. This is achieved by first generating as many random strings as there are characters in the alphabet, each string containing the respective alphabet character at a random position. The rest of the strings are randomly sampled without any further restrictions on the alphabet. These generated strings are then mixed together with the actual training examples by randomly shuffling all training examples, and both tasks are thus trained simultaneously. The random shuffling of training examples (i.e. individual words), and therefore breaking the semantic context, does not harm the training of our lemmatizer as it is anyway looking at individual words at a time. As in our training data the morphosyntactic tags are already included for each word, and the random autoencoder strings do not use any morphosyntactic tags, there is no requirement of running the tagger at training time, thus making the training data shuffling procedure straightforward. We chose to autoencode random strings rather than actual words as that way we do not need any external resources and the method is easily repeatable for any language.

### 7.2 Morphological transducers

In our second data augmentation method, we lean on additional morphological/lexical resources available for a particular language. In addition to UD, other projects are also striving to build unified morphological resources across many different languages. For example, the UniMorph project (Kirov *et al.* 2016) extracts and normalizes morphological paradigms from the Wiktionary free online dictionary site. Further, FSTs for morphological analysis and generation for a multitude of languages are available in the Apertium framework, which includes a pool of open source

resources for natural language processing (Tyers *et al.* 2010). Both UniMorph and Apertium frameworks can be used to collect inflected words and for each word a set of possible lemmas together with the corresponding morphological features. However, while these resources are unified within a project, their schema and guidelines differ from each other across different projects. For this reason, using a mixture of training examples gathered from two or more different sources is not a straightforward task. While harmonized annotations across different languages give a good starting point for multilingual conversion, the mapping is usually not fully deterministic (see, e.g., McCarthy *et al.* 2018 for detailed study of mapping from UD into UniMorph).

We expand our preliminary data augmentation experiments carried out during the CoNLL-18 ST, where we used the Apertium morphological transducers to collect additional training examples. A morphological transducer is a finite-state automaton including morphological paradigms (inflection regularities/rules) and a lexicographical database (lexicon), where each lexical entry (lemma) is assigned to the inflection paradigm it follows. These linguistic resources can be compiled into an efficient FST, an automaton which is able to return all matching lemmas and morphological hypotheses encoded in it for the given input word.

We set out to test whether improvements similar to those achieved with low-resource languages can also be seen with languages already including a reasonable amount of initial training data. We develop a language-agnostic feature mapping from Apertium features into UD, allowing us to cover all UD languages which have an Apertium morphological transducer available (Arabic, Armenian, Basque, Bulgarian, Buryat, Catalan, Czech, Danish, Dutch, English, Finnish, French, Galician, German, Greek, Hindi, Italian, Kazakh, Kurmanji, Latvian, Norwegian, Polish, Russian, Spanish, Swedish, Turkish, Ukrainian, and Urdu).

For each of these languages, we first gather a full vocabulary list sorted by word frequencies in descending order. These lists are gathered mainly from the web crawl data sets (Ginter *et al.* 2017), but for languages not included in the distributed web crawl data set (Armenian, Buryat, Kurmanji) we use Wikipedia dumps instead. The word frequency lists are then analyzed by the Apertium morphological transducers where for each unique word we obtain a set of possible lemmas and their corresponding morphological features. Words not recognized by the transducer (not part of the predefined lexicon) are simply discarded. All of these Apertium analyzes are then converted into the UD schema using our language-agnostic feature mapping where each morphological feature is converted into UD, based on a manually created look-up table. As the mapping from Apertium features into UD features is not a fully deterministic task, our language-agnostic feature mapping is designed for high precision and low recall, meaning that if a feature cannot be reliably translated, it will be dropped from the UD analyses. This approach may produce incomplete UD analyses, but we hypothesize that the lemmatizer model is robust enough to be able to utilize existing features without missing ones being too harmful for the training process, especially since in the actual training data these augmented examples are mixed together with the actual ones. The lemmas, on the other hand, we assume to be relatively harmonized between UD and Apertium by default, and these are used without any conversion or modification. After feature translation, we skip words which already appear in the original treebank training data, as well as all lemmas with a missing part-of-speech tag in the UD analysis due to an incomplete feature conversion, and all ambiguous words having two or more different lemmas with exactly the same morphological features. Finally, we pick a number of most common words from the UD converted and filtered transducer output, which are then mixed together with the original treebank training data. All training examples are randomly shuffled before training.

### 7.3 Data augmentation results

First, we compare the two augmentation methods against our basic system, where based on observations in Bergmanis *et al.* (2017), we mix 4000 additional training examples together with the original training data in both experiments. We decided to use a constant number of additional

**Table 1.** Evaluation of our two data augmentation methods, augmented with autoencoder and augmented with transducer as well as a mixed method, compared to our basic models. Additionally, we measure average percentage of words recognized by the transducer (Transducer Coverage) and average percentage of words having the correct lemma among the possible analyses (Transducer Recall), which represents an oracle accuracy achievable by transducers if all lemmas could be disambiguated correctly. All metrics are measured on token level, and in each column the highest accuracy value is bolded

Model	All treebanks	Excl. low resource	Transd. only treebanks
Basic	92.22	95.37	92.03
Augm. autoencoder 4K	92.89	95.42	93.11
Augm. transducer 4K	93.15	95.45	93.55
Augm. mixed 2K + 2K	93.12	95.47	93.45
Augm. mixed 4K + 4K	93.17	95.48	93.56
Augm. mixed 8K + 8K	<b>93.24</b>	<b>95.51</b>	<b>93.61</b>
Transd. coverage	–	–	86.76
Transd. recall	–	–	78.15

examples rather than a percentage to better account for the low-resource languages, the ones benefiting most from the experiment, where, for example, a 20% increase in training data would still translate to having less than 500 training examples. Second, we add experiments on using a mixture of both augmentation techniques and increasing the number of additional examples included. Additionally, we test how well a morphological transducer itself could serve as a lemmatizer by measuring its coverage (how many words from the test data are recognized by the transducer) and lemma recall (how many words from the test set have the correct lemma among the possible analyses given by the transducer). Lemma recall therefore gives an upper-bound, oracle accuracy achievable by the transducer, assuming that all lemmas in its output can be correctly disambiguated. Results are given in Table 1. We measure macro accuracy over all treebanks and results are given separately for three treebank groups: *All treebanks* includes all 76 treebanks studied in this paper, *Excluding low resource* is all treebanks except the 5 low-resource treebanks and *Transducer-only treebanks* is a set of 47 treebanks representing languages which have a morphological transducer available. Note that in *All treebanks* results the *Augm. transducer* row uses the basic model for treebanks where a transducer is not available, giving a realistic comparison against the *Augm. autoencoder* method which does not suffer from lacking resources. In the mixed experiments, if a transducer is not available for a language, the training data is enriched only with the autoencoder examples. The two direct transducer metrics (Transducer Coverage and Recall), however, can be realistically measured only for languages having a transducer available and the results reported for the *Transducer-only treebanks* group allow for a direct comparison between plain transducers and our models.

In all three groups, all augmentation methods are able to surpass the basic model, with the transducer-based method giving slightly better overall results than the autoencoder. When mixing the two methods, the same amount of total examples as in the plain transducer augmentation is divided evenly between the two methods. The mixed method is not able to surpass the transducer-based one, but when increasing the amount of additional mixed data, the performance also increases slightly, the mixed 8K + 8K, the largest mixed method tested, giving the best overall performance. When considering a macro-average over all treebanks, errors are reduced by 13%

relative compared to our basic models. However, when excluding the five low resource treebanks already discussed in Section 6, the difference is smaller, and the relative error reduction becomes a mere 3%, demonstrating that—unsurprisingly—most of the benefit comes from the low-resource languages and only a minimal improvement can be seen with reasonably sized training data sets.

The average coverage for the morphological transducers is 86%, with recall being 78%. These numbers are clearly below our lemmatization methods, showing that, averaged across many languages, the approach relying on a predefined lexicon and ruleset does not fare favorably to sequence-to-sequence machine learning methods. The average transducer coverage is on par with the one reported by Tyers *et al.* (2010), where coverage numbers reported for a set of languages varies between 80% and 98%; however, with our set of languages, the variation is much higher ranging between 5% and 99%, and clearly the transducers in the lower coverage region are missing much of the core vocabulary. These are measured without using morphological guessers, where unknown words can be analyzed based only on their morphological shape (e.g. known suffixes). However, as the guessers consider every possible mapping allowed by the rules of the language, in many cases a great number of different alternatives is returned, which would need to be disambiguated later on. We therefore leave it as a future work to study whether morphological guessers and sequence-to-sequence lemmatizers can have a shared interest. By comparing the transducer coverage and recall, we can have an estimate of how harmonized the lemmas are between Apertium transducers and UD treebanks on average. If 86% of words are recognized by the transducer, but only 78% are having a “correct” lemma analysis, then 8% of the treebank words are recognized but with a “wrong” lemma, hinting at an incompatible analysis. We leave it as a future study to examine, whether the differences are systematic and further gains could be obtained with filtering or harmonizing the lemma annotations between Apertium and UD in addition to harmonizing morphological features. Such a study however requires the knowledge of each of the involved languages.

## 8. Discussion

### 8.1 Result summary

In Table 2, we summarize the results of all the major experiments reported in this paper. For each treebank, we present the accuracy of our best overall method, *Augm. mixed 8K + 8K*, and for comparison, we also add results for our basic method as well as the best overall baseline method, *UDPipe Future*. The comparison of our system and the UDPipe Future baseline is visualized by coloring each line green where our Mixed 8K+8K method is better than the UDPipe Future baseline. As discussed in Section 5.3, all numbers are measured on top of predicted segmentation, therefore reflecting a realistic expectation of the performance with no gold-standard data used at any point during prediction.

Out of the 76 treebanks, our method outperforms the UDPipe Future baseline on 62 treebanks. On average, across the 76 treebanks, this translates to a relative 19% error reduction. On 36 treebanks the relative error reduction is more than 20%, meaning that we are able to remove at least one fifth of the errors the best baseline system is making.

While the autoencoding augmentation method does not require any additional data, the transducer-based techniques move the system into an unconstrained setting, if considering a task setup where only the given treebanks are allowed in system training. However, in real-life situations, where all available data are allowed, the comparison between our augmented system and the baseline systems is fair. Such a real-life task setting was used, for example, in the CoNLL 2018 and 2017 multilingual parsing shared tasks, where a list of additional resources apart from the treebanks were given to all task participants. These allowed resources also included the Apertium morphological transducers, which makes the comparison between our augmentation methods and baseline systems from the CoNLL 2018 shared task fair. The difference between our system



**Table 2.** Lemmatization accuracies for all 76 treebanks studied in this paper measured on test data with predicted segmentation. Green color indicates treebanks where our overall best method, Augm. Mixed 8K + 8K, outperforms the best overall baseline, UDPipe Future

Treebank	Treebank category	UDPipe future	Our basic	Our augm. mixed	Relative diff UDP-Ours (%)
Afrikaans-AfriBooms	big	97.11	97.59	97.76	22.5
Ancient Greek-PROIEL	big	91.08	97.27	97.31	69.8
Ancient Greek-Perseus	big	81.78	89.40	89.60	42.9
Arabic-PADT	big	88.94	89.47	89.46	4.7
Armenian-ArmTDP	low	57.46	66.82	71.81	33.7
Basque-BDT	big	95.19	96.66	96.81	33.7
Bulgarian-BTB	big	97.41	98.21	98.17	29.3
Buryat-BDT	low	56.83	25.55	56.05	-1.8
Catalan-AnCora	big	98.90	97.57	97.66	-53.0
Chinese-GSD	big	90.01	87.74	89.55	-4.4
Croatian-SET	big	96.69	96.81	96.87	5.4
Czech-CAC	big	98.14	98.19	98.34	10.8
Czech-FicTree	big	97.80	98.74	98.84	47.3
Czech-PDT	big	98.71	98.48	98.52	-12.8
Czech-PUD	PUD	96.44	96.04	96.14	-7.8
Danish-DDT	big	96.66	97.79	97.88	36.5
Dutch-Alpino	big	96.76	96.67	96.85	2.8
Dutch-LassySmall	big	95.78	97.40	97.44	39.3
English-EWT	big	97.23	96.96	96.94	-9.5
English-GUM	big	96.18	96.07	96.21	0.8
English-LinES	big	96.44	96.54	96.79	9.8
English-PUD	PUD	95.87	96.39	96.40	12.8
Estonian-EDT	big	94.88	96.56	96.60	33.6
Finnish-FTB	big	94.74	97.02	97.18	46.4
Finnish-PUD	PUD	90.64	95.05	95.13	48.0
Finnish-TDT	big	90.18	95.24	95.40	53.2
French-GSD	big	96.75	96.90	96.91	4.9
French-Sequoia	big	97.36	97.98	98.06	26.5
French-Spoken	big	95.98	96.77	97.04	26.4
Galician-CTG	big	97.53	97.88	97.92	15.8
Galician-TreeGal	small	95.05	94.98	95.50	9.1

Table 2. Continued

Treebank	Treebank category	UDPipe future	Our basic	Our augm. mixed	Relative diff UDP-Ours (%)
German-GSD	big	96.14	96.68	96.56	10.9
Gothic-PROIEL	big	92.39	96.10	96.21	50.2
Greek-GDT	big	94.74	97.22	97.26	47.9
Hebrew-HTB	big	82.88	82.90	82.93	0.3
Hindi-HDTB	big	98.45	98.68	98.70	16.1
Hungarian-Szeged	big	92.99	94.53	94.57	22.5
Indonesian-GSD	big	99.60	99.69	99.68	20.0
Irish-IDT	small	87.52	90.62	90.52	24.0
Italian-ISDT	big	98.21	98.09	98.16	-2.7
Italian-PoSTWITA	big	94.91	96.61	96.63	33.8
Japanese-GSD	big	90.01	89.94	89.63	-3.7
Kazakh-KTB	low	57.36	48.61	57.43	0.2
Korean-GSD	big	91.37	93.83	93.94	29.8
Korean-Kaist	big	93.53	94.38	94.39	13.3
Kurmanji-MG	low	52.44	42.45	64.83	26.1
Latin-ITTB	big	98.56	98.66	98.67	7.6
Latin-PROIEL	big	95.54	97.14	97.20	37.2
Latin-Perseus	small	75.44	85.37	85.27	40.0
Latvian-LVTB	big	93.33	93.69	93.95	9.3
North Sami-Giella	small	78.43	89.54	89.70	52.2
Norwegian-Bokmaal	big	98.20	97.87	97.97	-11.3
Norwegian-Nynorsk	big	97.80	97.71	97.72	-3.5
Norwegian-NynorskLIA	small	92.65	92.91	94.51	25.3
Old Church Slavonic-PROIEL	big	88.93	95.33	95.14	56.1
Persian-Seraji	big	97.05	96.99	96.77	-8.7
Polish-LFG	big	96.73	97.50	97.66	28.4
Polish-SZ	big	95.31	96.93	97.08	37.7
Portuguese-Bosque	big	97.38	97.53	97.58	7.6
Romanian-RRT	big	97.61	98.25	98.23	25.9
Russian-SynTagRus	big	97.94	98.16	98.15	10.2
Russian-Taiga	small	83.55	88.47	89.32	35.1
Serbian-SET	big	96.56	97.09	97.17	17.7

Table 2. Continued

	Treebank	UDPipe	Our	Our augm.	Relative diff
Treebank	category	future	basic	mixed	UDP-Ours (%)
Slovak-SNK	big	95.66	96.27	96.35	15.9
Slovenian-SSJ	big	96.22	96.35	96.49	7.1
Slovenian-SST	small	92.56	95.06	94.90	31.5
Spanish-AnCora	big	99.02	98.45	98.48	-35.5
Swedish-LinES	big	96.61	96.87	97.29	20.1
Swedish-PUD	PUD	86.23	86.69	87.47	9.0
Swedish-Talbanken	big	97.08	97.81	97.98	30.8
Turkish-IMST	big	92.74	94.85	95.16	33.3
Ukrainian-IU	big	95.94	96.52	96.62	16.7
Upper Sorbian-UFAL	low	63.54	53.73	54.80	-19.3
Urdu-UDTB	big	97.33	97.42	97.43	3.7
Uyghur-UDT	big	92.86	94.09	94.15	18.1
Vietnamese-VTB	big	84.76	84.16	84.26	-3.2
Average		91.64	92.22	93.24	19.1%

and a standard context-based lemmatization system is that integrating information from these additional sources is much easier with our task setting where the lemmatizer does not need the words to appear in a natural context.

## 8.2 Generalization and error propagation

To understand the generalization capability of the lemmatizer when the segmentation and morphological tagging effects are disregarded, we compare the lemmatization accuracy on top of predicted segmentation to gold-standard segmentation (sentence and word level), as well as on top of predicted morphosyntactic features to gold-standard morphosyntactic features. The same experiment also measures the risk of error propagation, where the lemmatizer makes a mistake due to incorrectly predicted morphosyntactic features. Results for all treebanks are available in Appendix A. When comparing the lemmatization accuracy of the five low-resource languages (Armenian, Buryat, Kazakh, Kurmanji, Upper Sorbian) on predicted and gold morphosyntactic features, the four transducer languages (Armenian, Buryat, Kazakh, Kurmanji) appear to generalize extremely well, gold morphosyntactic features increasing the accuracy from 58%–74% to 91%–96%. For Upper Sorbian, the one low-resource language without a transducer, the generalization ability is clearly worse, gold tags increasing accuracy only from 55% to 74%. These results suggest that the data augmentation techniques utilizing a morphological transducer are sufficient enough to train a high-quality lemmatizer if reliable morphosyntactic features are available. However, at the same time it shows that in extreme cases where the accuracy of part-of-speech tagging is barely above 50%, errors from the tagger component propagate notably. As a future work, it would be interesting to study whether morphological transducers could be used to create artificial data for context-dependent morphological tagging so as to improve the tagger performance as well.

Currently, the lemmatizer is the last component in the parsing pipeline, thus not affecting the labeled attachment score of the syntactic parser. The parser currently used in the pipeline was originally designed to not consider lemmas at all; however, the lemmatizer component could be located before the syntactic parser as well, making it possible to establish whether using lemmas as additional features during parsing would improve its performance.

### 8.3 Future work

We acknowledge that the morphological transducers used in our data augmentation study may not have been utilized to their full power. Our straightforward feature mapping from the Apertium framework into UD was designed to be language agnostic, thus suffering from inconsistencies in annotations between different languages and treebanks. A more focused attempt on a particular, well-chosen language with an improved morphological transducer, language-specific conversion or detailed parameter tuning could yield better results. While Apertium can be considered a trustworthy source for unified morphological resources, for many languages, more developed language-specific transducers exist. For example, if particularly working on Turkish, Finnish, or Hungarian, one should consider using morphological transducers by Çöltekin (2010), Pirinen (2015), and Trón *et al.* (2006). A focused per-language effort is naturally entirely out of scope of this current work, which can nevertheless serve as a basis for such a language-specific development. Similar argumentation is suggested by Pirinen (2019), who carried out a focused evaluation of our lemmatization system and the OMorFi morphological analyzer (Pirinen 2015) on the Finnish language. OMorFi is a mature system, being the result of a major development effort spanning over several years. Its output is in the UD scheme, providing a valid point of comparison. A lemmatization performance of our pipeline far superior to that of OMorFi is reported, leading to the conclusion that the machine learning approach is indeed highly competitive with the traditional transducers and can be seen as the preferred approach to developing lemmatizers for new languages. However, we leave it as a future work to study whether combining such a morphological transducer and machine learning approach in a targeted data augmentation effort would yield higher improvements for lemmatization accuracy than presented in this paper.

Another interesting direction to expand the work in future would be to test how well the lemmatizer works on short text segments, for example, with search queries, where deep learning systems traditionally need to be trained separately to match the different style of writing, for example, very often omitting the main verb. As the lemmatizer is operating on the word level without a notion of context, this should not pose an issue during the lemmatization. However, a separate question is how reliable a morphological tagger would be with such short text segments.

### 8.4 Model and software release

We release trained models for all 76 treebanks experimented in this paper, embedded into a full parsing pipeline including segmentation, tagging, syntactic parsing, and lemmatization. The parsing pipeline source code is available at <https://turkunlp.org/Turku-neural-parser-pipeline> under the Apache 2.0 license. It includes trained models for all the necessary components (segmentation, tagging, syntactic parsing, and lemmatization), trained on the UD v2.2 treebanks. The whole processing pipeline can be executed with a single command, removing the need for data reformatting between the different analysis components. The pipeline runs in a Python environment which can be installed with or without GPU support. To increase the usability across different platforms, we also provide a publicly accessible Docker image, which wraps the pipeline in a container which can be executed without manual installation, assuring that the pipeline can be executed and the results replicated also in the future.

### 8.5 Training and prediction speed

Typical training times for the lemmatizer models on UD treebanks with 50 training epochs are 1–2 hours on one Nvidia GeForce K80 GPU card. The largest treebanks (Czech-PDT 1.2M tokens and Russian-SynTagRus 870K tokens) took approximately 15 hours to train for the full 50 epochs. However, the training usually converges between epochs 30 and 40, and therefore, training time could be reduced using an early stopping criterion.

In prediction time, we present several advantages over previous sequence-to-sequence lemmatizer models. First, by using morphosyntactic features instead of a sliding window of text to represent the contextual information, after running the context-dependent morphological tagger, the lemmatizer is able to process each word independently from its textual sentence context, and therefore we only need to lemmatize each unique word and feature combination. This enables us to (1) only lemmatize unique items inside each textual batch and (2) store a cache of common pre-analyzed words, and only run the sequence-to-sequence model for words not already present in this global lemma cache. Together with the trained models, we distribute such a global cache file for each language.

Prediction times for the full parsing pipeline, including segmentation, tagging, syntactic parsing, and lemmatization, are on the order of 1300 tokens per second (about 100 sentences per second) on an Nvidia GeForce GTX 1070 card. On a server-grade CPU-only computer (24 cores and 250GB RAM), prediction times are 350 tokens per second, while on a consumer CPU-only laptop (8 cores and 8GB of RAM), the full pipeline can process about 280 tokens per second. These are measured with a pre-analyzed lemma cache collected from the training data, and prediction times especially on CPU could be yet improved by collecting a larger pre-analyzed lemma cache using, for example, large web corpora.

## 9. Conclusions

In this paper, we have introduced a novel sequence-to-sequence lemmatization method utilizing morphosyntactic tags to inform the model about the context of the word. We validated the hypothesis that the tags provide a sufficient disambiguation context using statistics from the UD treebanks across a large number of languages. We presented a careful evaluation of our method over several baselines and 52 different languages showing that the method surpasses all the baseline systems, reducing relative errors on average by 19% across 76 treebanks compared to the best overall baseline. The lemmatizer presented in this work was also used as our entry in the CoNLL-18 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies, where we achieved the 1st place out of 26 teams on two evaluation metrics incorporating lemmatization. Additionally, we investigated two different data augmentation methods to boost the lemmatization performance of our base system. We found that augmenting the training data using a mixture of autoencoder training and the output of a morphological transducer decreases the error rate by 13% relative to the unaugmented system, with the gain being unsurprisingly concentrated on the low-resource languages.

As an overall conclusion, we have demonstrated a highly competitive performance of the generic sequence-to-sequence paradigm on the lemmatization task, surpassing in accuracy prior methods specifically developed for lemmatization.

The lemmatization models for all languages reported in the paper, source code, and materials for all experiments, the full parsing pipeline source code, and parsing models, as well as an easy-to-use Docker container, are available at <https://turkunlp.org/Turku-neural-parser-pipeline> under the Apache 2.0 license.

**Acknowledgments.** We gratefully acknowledge the support of Academy of Finland, CSC – IT Center for Science, and the NVIDIA Corporation GPU Grant Program.

## References

- Aker A., Petrak J. and Sabbah F. (2017). An extensible multilingual open source lemmatizer. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, Varna, Bulgaria. INCOMA Ltd., pp. 40–45.
- Bergmanis T. and Goldwater S. (2018). Context sensitive neural lemmatization with Lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, vol. 1. Association for Computational Linguistics, pp. 1391–1400.
- Bergmanis T., Kann K., Schütze H. and Goldwater S. (2017). Training data augmentation for low-resource morphological inflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pp. 31–39.
- Chakrabarty A., Pandit O.A. and Garain U. (2017). Context sensitive lemmatization using two successive bidirectional gated recurrent networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, vol. 1. Association for Computational Linguistics, pp. 1481–1491.
- Chrupala G., Dinu G. and Van Genabith J. (2008). Learning morphology with Morfette. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA), pp. 2362–2367.
- Çöltekin Ç. (2010). A freely available morphological analyzer for Turkish. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, vol. 2. European Language Resources Association (ELRA), pp. 19–28.
- Costa A. and Nannicini G. (2018). RBFOpt: an open-source library for black-box optimization with costly function evaluations. *Mathematical Programming Computation* **10**, 597–629.
- Cotterell R., Kirov C., Sylak-Glassman J., Walther G., Vylomova E., Xia P., Faruqui M., Kübler S., Yarowsky D., Eisner J. and Hulden M. (2017). CoNLL-SIGMORPHON 2017 Shared Task: universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, Vancouver, Canada. Association for Computational Linguistics, pp. 1–30.
- Dozat T. and Manning C.D. (2017). Deep biaffine attention for neural dependency parsing. In *Proceedings of the 2017 International Conference on Learning Representations (ICLR'17)*.
- Dozat T., Qi P. and Manning C.D. (2017). Stanford's graph-based neural dependency parser at the CoNLL 2017 Shared Task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, pp. 20–30.
- Ginter F., Hajič J., Luotolahti J., Straka M. and Zeman D. (2017). CoNLL 2017 Shared Task - automatically annotated raw texts and word embeddings. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Jeon S., Cho K., Memisevic R. and Bengio Y. (2015). On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, Beijing, China. Association for Computational Linguistics, pp. 1–10.
- Kanerva J., Ginter F., Miekka N., Leino A. and Salakoski T. (2018). Turku neural parser pipeline: an end-to-end system for the CoNLL 2018 Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium. Association for Computational Linguistics.
- Kann K. and Schütze H. (2017a). The LMU system for the CoNLL-SIGMORPHON 2017 shared task on universal morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, Vancouver, Canada. Association for Computational Linguistics, pp. 40–48.
- Kann K. and Schütze H. (2017b). Unlabeled data for morphological generation with character-based sequence-to-sequence models. In *Proceedings of the 1st Workshop on Subword and Character Level Models in NLP (SCLeM 2017)*, Copenhagen, Denmark. Association for Computational Linguistics.
- Karttunen L. and Beesley K.R. (1992). *Two-level rule compiler*. Xerox Corporation. Palo Alto Research Center.
- Kingma D. and Ba J. (2015). Adam: a method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.
- Kirov C., Sylak-Glassman J., Que R. and Yarowsky D. (2016). Very-large scale parsing and normalization of Wiktionary morphological paradigms. In Calzolari N. (Conference Chair), Choukri K., Declerck T., Goggi S., Grobelnik M., Maegaard B., Mariani J., Mazzo H., Moreno A., Odijk J. and Piperidis S. (eds), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Klein G., Kim Y., Deng Y., Senellart J. and Rush A.M. (2017). OpenNMT: open-source toolkit for neural machine translation. In *Proceedings of the 55th annual meeting of the Association for Computational Linguistics (ACL'17)*, Vancouver, Canada. Association for Computational Linguistics.
- Kondratyuk D., Gavenčiak T., Straka M. and Hajič J. (2018). LemmaTag: jointly tagging and lemmatizing for morphologically rich languages with BRNNs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 4921–4928.
- Koskenniemi K. (1984). A general computational model for word-form recognition and production. In *Proceedings of the 10th International Conference on Computational Linguistics*, USA. Association for Computational Linguistics, pp. 178–181.

- Liu L. and Hulden M.** (2017). Evaluation of finite state morphological analyzers based on paradigm extraction from Wiktionary. In *Proceedings of the 13th International Conference on Finite State Methods and Natural Language Processing (FSM/NLP 2017)*, Umeå, Sweden. Association for Computational Linguistics, pp. 69–74.
- Luong M.-T., Sutskever I., Le Q.V., Vinyals O. and Zaremba W.** (2015a). Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, Beijing, China. Association for Computational Linguistics, pp. 11–19.
- Luong T., Pham H. and Manning C.D.** (2015b). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal. Association for Computational Linguistics, pp. 1412–1421.
- McCarthy A.D., Silfverberg M., Cotterell R., Hulden M. and Yarowsky D.** (2018). Marrying universal dependencies and universal morphology. In *Proceedings of the 2018 Workshop on Universal Dependencies (UDW 2018)*. Association for Computational Linguistics.
- Müller T., Cotterell R., Fraser A. and Schütze H.** (2015). Joint lemmatization and morphological tagging with Lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal. Association for Computational Linguistics, pp. 2268–2274.
- Nivre J., Abrams M., Agić Ž., Ahrenberg L., Antonsen L., Aranzabe M.J., Arutie G., Asahara M., Ateyah L., Attia M., Atutxa A., Augustinus L., Badmaeva E., Ballesteros M., Banerjee E., Bank S., Barbu Mititelu V., Bauer J., Bellato S., Bengoetxea K., Bhat R.A., Biagetti E., Bick E., Blokland R., Bobicev V., Börstell C., Bosco C., Bouma G., Bowman S., Boyd A., Burchardt A., Candito M., Caron B., Caron G., Cebiroğlu Eryiğit G., Celano G.G.A., Cetin S., Chalub F., Choi J., Cho Y., Chun J., Cinková S., Collomb A., Çöltekin Ç., Connor M., Courtin M., Davidson E., de Marneffe M.-C., de Paiva V., de Ilarraza A.D., Dickerson C., Dirix P., Dobrovoljc K., Dozat T., Droganova K., Dwivedi P., Eli M., Elkahky A., Ephrem B., Erjavec T., Etienne A., Farkas R., Fernandez Alcalde H., Foster J., Freitas C., Gajdošová K., Galbraith D., Garcia M., Gärdenfors M., Gerdes K., Ginter F., Goenaga I., Gojenaga K., Gökrmak M., Goldberg Y., Gómez Guinovart X., Gonzáles Saavedra B., Grioni M., Grūztis N., Guillaume B., Guillot-Barbance C., Habash N., Hajič J., Hajič jr. J., Hà MÓ L., Han N.-R., Harris K., Haug D., Hladká B., Hlaváčová J., Hociung F., Hohle P., Hwang J., Ion R., Irimia E., Jelnek T., Johannsen A., Jørgensen F., Kaskara H., Kahane S., Kanayama H., Kanerva J., Kayadelen T., Kettnerová V., Kirchner J., Kotsyba N., Krek S., Kwak S., Laippala V., Lambertino L., Lando T., Larasati S.D., Lavrentiev A., Lee J., Lê H'ông P., Lenci A., Lertpradit S., Leung H., Li C.Y., Li J., Li K., Lim K.T., Ljubešić N., Loginova O., Lyashevskaya O., Lynn T., Macketanz V., Makazhanov A., Mandl M., Manning C., Manurung R., Mărănduc C., Mareček D., Marheinecke K., Martinez Alonso H., Martins A., Mašek J., Matsumoto Y., McDonald R., Mendonça G., Miekka N., Missilä A., Mititelu C., Miyao Y., Montemagni S., More A., Moreno Romero L., Mori S., Mortensen B., Moskalevskiy B., Muischnek K., Murawaki Y., Müürisep K., Nainwani P., Navarro Horňáček J.I., Nedoluzhko A., Nešpore-Běrzkalne G., Nguy~ên Thi L., Nguy~ên Thi Minh H., Nikolaev V., Nitisaroj R., Nurmi H., Ojala S., Olúòkun A., Omura M., Osenova P., Östling R., Øvrelid L., Partanen N., Pascual E., Passarotti M., Patejuk A., Peng S., Perez C.-A., Perrier G., Petrov S., Piitulainen J., Pitler E., Plank B., Poibeau T., Popel M., Pretkálnina L., Prévost S., Prokopidis P., Przepiórkowski A., Puolakainen T., Pyysalo S., Rääbis A., Rademaker A., Ramasamy L., Rama T., Ramisch C., Ravishankar V., Real L., Reddy S., Rehm G., Rießler M., Rinaldi L., Rituma L., Rocha L., Romanenko M., Rosa R., Rovati D., Rosca V., Rudina O., Sadde S., Saleh S., Samardžić T., Samson S., Sanguinetti M., Saulte B., Sawanakunanon Y., Schneider N., Schuster S., Seddah D., Seeker W., Seraji M., Shen M., Shimada A., Shohibussirri M., Sichinava D., Silveira N., Simi M., Simionescu R., Simkó K., Šimková M., Simov K., Smith A., Soares-Bastos I., Stella A., Straka M., Strnadová J., Suhr A., Sulubacak U., Szántó Z., Taji D., Takahashi Y., Tanaka T., Tellier I., Trosterud T., Trukhina A., Tsarfaty R., Tyers F., Uematsu S., Urešová Z., Uria L., Uszkoreit H., Vajjala S., van Niekerk D., van Noord G., Varga V., Vincze V., Wallin L., Washington J.N., Williams S., Wirén M., Woldemariam T., Wong T.-s., Yan C., Yavrumyan M.M., Yu Z., Žabokrtský Z., Zeldes A., Zeman D., Zhang M. and Zhu H. (2018). Universal Dependencies 2.2. LIN DAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.**
- Nivre J., de Marneffe M.-C., Ginter F., Goldberg Y., Hajič J., Manning C.D., McDonald R.T., Petrov S., Pyysalo S., Silveira N., et al.** (2016). Universal Dependencies v1: a multilingual treebank collection. In *Proceedings of Language Resources and Evaluation Conference (LREC'16)*, Portorož, Slovenia. European Language Resources Association (ELRA).
- Östling R. and Bjerva J.** (2017). SU-RUG at the CoNLL-SIGMORPHON 2017 Shared Task: morphological inflection with attentional sequence-to-sequence models. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, Vancouver, Canada. Association for Computational Linguistics.
- Pirinen T.A.** (2015). Development and use of computational morphology of Finnish in the open source and open science era: notes on experiences with OMorFi development. *SKY Journal of Linguistics* 28, 381–393.
- Pirinen T.A.** (2019). Neural and rule-based Finnish NLP models—expectations, experiments and experiences. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, Tartu, Estonia. Association for Computational Linguistics, pp. 104–114.
- Qi P., Dozat T., Zhang Y. and Manning C.D.** (2018). Universal Dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium. Association for Computational Linguistics, pp. 160–170.

- Sennrich R., Haddow B. and Birch A.** (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54rd Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany. Association for Computational Linguistics, pp. 1715–1725.
- Smith N.A., Smith D.A. and Tromble R.W.** (2005). Context-based morphological disambiguation with random fields. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, Canada. Association for Computational Linguistics, pp. 475–482.
- Straka M.** (2018a). CoNLL 2018 Shared Task - UDPipe baseline models and supplementary materials. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Straka M.** (2018b). UDPipe 2.0 prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium. Association for Computational Linguistics, pp. 197–207.
- Straka M., Hajič J. and Straková J.** (2016). UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia. European Language Resources Association (ELRA).
- Trón V., Halácsy P., Rebrus P., Rung A., Vajda P. and Simon E.** (2006). Morphdb.hu: Hungarian lexical database and morphological grammar. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA), pp. 1670–1673.
- Tyers F., Sánchez-Martínez F., Ortiz-Rojas S. and Forcada M.** (2010). Free/open-source resources in the Apertium platform for machine translation research and development. *The Prague Bulletin of Mathematical Linguistics* **93**, 67–76.
- Zeman D., Hajič J., Popel M., Potthast M., Straka M., Ginter F., Nivre J. and Petrov S.** (2018). CoNLL 2018 Shared Task: multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium. Association for Computational Linguistics, pp. 1–20.
- Zeman D., Popel M., Straka M., Hajič J., Nivre J., Ginter F., Luotolahti J., Pyysalo S., Petrov S., Potthast M., Tyers F., Badmaeva E., Gökrımak M., Nedoluzhko A., Cinková S., Hajič jr. J., Hlaváčová J., Kettnerová V., Urešová Z., Kanerva J., Ojala S., Missilä A., Manning C., Schuster C., Reddy S., Taji D., Habash N., Leung H., de Marneffe M.-C., Sanguinetti M., Simi M., Kanayama H., de Paiva V., Drogonova K., Martínez Alonso H., Uszkoreit H., Macketanz V., Burchardt A., Harris K., Marheinecke K., Rehm G., Kayadelen T., Attia M., Elkahky A., Yu Z., Pitler E., Lertpradit S., Mandl M., Kirchner J., Fernandez Alcalde H., Strnadova J., Banerjee E., Manurung R., Stella A., Shimada A., Kwak S., Mendonça G., Lando T., Nitisaroj R. and Li J.** (2017). CoNLL 2017 Shared Task: multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Vancouver, Canada. Association for Computational Linguistics.

## A Lemmatization accuracy with gold segmentation and morphology

**Table A1.** Lemmatization accuracy for all treebanks measured on gold and predicted segmentation and tagging

Treebank		Tokens	Sents	UPOS	XPOS	UFeats	Lemmas	LAS
Afrikaans-AfriBooms	raw text	99.75	98.25	97.32	93.67	96.71	97.76	85.14
	gold seg	–	–	97.55	93.85	96.92	97.95	85.67
	gold seg+mor	–	–	–	–	–	98.63	88.31
Ancient Greek-PROIEL	raw text	100.00	44.57	97.00	97.17	91.86	97.31	75.88
	gold seg	–	–	97.06	97.29	91.93	97.43	82.12
	gold seg+mor	–	–	–	–	–	98.87	84.96
Ancient Greek-Perseus	raw text	99.96	98.73	91.93	83.87	89.94	89.60	73.26
	gold seg	–	–	92.01	83.95	89.99	89.61	73.42
	gold seg+mor	–	–	–	–	–	92.73	77.63



**Table A1.** Continued

Treebank		Tokens	Sents	UPOS	XPOS	UFeats	Lemmas	LAS
Arabic-PADT	raw text	99.98	80.89	90.47	87.37	87.56	89.46	72.44
	gold seg	-	-	96.75	93.73	93.95	95.62	82.46
	gold seg+mor	-	-	-	-	-	98.34	84.33
Armenian-ArmTDP	raw text	97.21	92.41	69.31	96.47	46.05	71.81	29.74
	gold seg	-	-	71.21	100.00	47.71	73.77	30.90
	gold seg+mor	-	-	-	-	-	93.47	37.00
Basque-BDT	raw text	99.96	99.08	96.01	99.96	92.07	96.81	82.38
	gold seg	-	-	96.06	100.00	92.10	96.86	82.52
	gold seg+mor	-	-	-	-	-	98.50	86.08
Bulgarian-BTB	raw text	99.92	92.85	98.61	96.41	97.41	98.17	89.60
	gold seg	-	-	98.70	96.50	97.50	98.25	90.54
	gold seg+mor	-	-	-	-	-	99.42	91.59
Buryat-BDT	raw text	97.07	90.90	37.95	97.07	35.42	56.05	13.13
	gold seg	-	-	39.32	100.00	37.24	58.11	13.43
	gold seg+mor	-	-	-	-	-	90.99	15.35
Catalan-AnCora	raw text	99.97	99.03	97.32	97.34	96.66	97.66	90.04
	gold seg	-	-	97.36	97.37	96.70	97.69	90.15
	gold seg+mor	-	-	-	-	-	99.24	92.30
Chinese-GSD	raw text	89.55	98.20	85.83	85.60	88.74	89.55	66.26
	gold seg	-	-	95.28	95.00	99.10	100.00	81.38
	gold seg+mor	-	-	-	-	-	100.00	86.61
Croatian-SET	raw text	99.92	95.36	98.02	99.92	91.91	96.87	86.19
	gold seg	-	-	98.09	100.00	91.94	96.94	86.65
	gold seg+mor	-	-	-	-	-	98.37	87.78
Czech-CAC	raw text	99.97	99.76	98.96	95.37	94.65	98.34	90.52
	gold seg	-	-	99.00	95.41	94.69	98.39	90.59
	gold seg+mor	-	-	-	-	-	99.48	91.72
Czech-FicTree	raw text	99.97	98.37	98.51	94.56	95.29	98.84	90.63
	gold seg	-	-	98.53	94.59	95.33	98.87	90.83
	gold seg+mor	-	-	-	-	-	99.63	92.39
Czech-PDT	raw text	99.93	92.29	98.72	95.41	95.18	98.52	90.54
	gold seg	-	-	98.80	95.54	95.30	98.60	91.48
	gold seg+mor	-	-	-	-	-	99.68	92.42

Table A1. Continued

Treebank		Tokens	Sents	UPOS	XPOS	UFeats	Lemmas	LAS
Czech-PUD	raw text	99.28	95.40	96.10	92.19	92.11	96.14	84.71
	gold seg	-	-	96.51	92.62	92.57	96.54	85.58
	gold seg+mor	-	-	-	-	-	97.52	87.20
Danish-DDT	raw text	99.87	87.96	97.31	99.87	97.10	97.88	82.96
	gold seg	-	-	97.47	100.00	97.25	98.01	84.34
	gold seg+mor	-	-	-	-	-	99.20	86.75
Dutch-Alpino	raw text	99.83	90.80	96.13	94.44	96.44	96.85	85.36
	gold seg	-	-	96.32	94.65	96.61	97.04	86.56
	gold seg+mor	-	-	-	-	-	98.61	89.41
Dutch-LassySmall	raw text	99.82	72.23	95.87	94.22	95.66	97.44	81.31
	gold seg	-	-	96.08	94.59	96.03	97.68	84.95
	gold seg+mor	-	-	-	-	-	99.39	88.42
English-EWT	raw text	99.03	75.33	94.85	94.64	95.95	96.94	82.67
	gold seg	-	-	95.77	95.63	96.90	97.81	86.93
	gold seg+mor	-	-	-	-	-	99.73	89.57
English-GUM	raw text	99.75	78.79	93.37	93.28	95.69	96.21	80.67
	gold seg	-	-	93.64	93.56	95.99	96.39	83.00
	gold seg+mor	-	-	-	-	-	98.08	86.00
English-LinES	raw text	99.95	88.08	96.43	95.04	96.74	96.79	79.44
	gold seg	-	-	96.49	95.13	96.79	96.83	80.09
	gold seg+mor	-	-	-	-	-	97.19	81.66
English-PUD	raw text	99.74	95.57	94.90	94.20	95.07	96.40	85.40
	gold seg	-	-	95.13	94.44	95.33	96.65	86.06
	gold seg+mor	-	-	-	-	-	98.94	88.23
Estonian-EDT	raw text	99.91	90.02	96.43	97.87	95.71	96.60	84.14
	gold seg	-	-	96.49	97.94	95.78	96.69	85.09
	gold seg+mor	-	-	-	-	-	98.34	87.92
Finnish-FTB	raw text	100.00	87.04	96.18	95.15	96.45	97.18	86.99
	gold seg	-	-	96.29	95.23	96.55	97.22	88.92
	gold seg+mor	-	-	-	-	-	99.19	92.34
Finnish-PUD	raw text	99.63	92.20	96.91	0.00	96.72	95.13	88.88
	gold seg	-	-	97.23	0.00	97.03	95.43	89.23
	gold seg+mor	-	-	-	-	-	96.67	88.92

Table A1. Continued

Treebank		Tokens	Sents	UPOS	XPOS	UFeats	Lemmas	LAS
Finnish-TDT	raw text	99.69	86.75	96.57	97.61	95.41	95.40	86.48
	gold seg	–	–	96.92	97.89	95.69	95.70	88.37
	gold seg+mor	–	–	–	–	–	97.68	90.84
French-GSD	raw text	99.66	92.12	95.96	98.78	95.73	96.91	85.62
	gold seg	–	–	97.15	100.00	96.86	98.11	87.58
	gold seg+mor	–	–	–	–	–	99.38	88.81
French-Sequoia	raw text	99.79	82.77	97.42	99.09	97.04	98.06	87.42
	gold seg	–	–	98.44	100.00	97.97	99.00	89.96
	gold seg+mor	–	–	–	–	–	99.66	91.19
French-Spoken	raw text	100.00	21.63	94.72	97.51	100.00	97.04	69.15
	gold seg	–	–	94.81	97.49	100.00	97.21	76.18
	gold seg+mor	–	–	–	–	–	97.95	78.58
Galician-CTG	raw text	99.84	96.59	97.07	96.87	98.96	97.92	81.64
	gold seg	–	–	97.85	97.67	99.78	98.73	83.25
	gold seg+mor	–	–	–	–	–	99.42	85.22
Galician-TreeGal	raw text	99.69	83.90	93.81	90.97	92.92	95.50	72.88
	gold seg	–	–	94.95	91.92	93.96	96.53	76.04
	gold seg+mor	–	–	–	–	–	98.02	79.42
German-GSD	raw text	99.58	81.32	93.81	96.56	90.20	96.56	78.64
	gold seg	–	–	94.11	97.02	90.87	96.95	80.72
	gold seg+mor	–	–	–	–	–	97.28	82.82
Gothic-PROIEL	raw text	100.00	28.03	95.49	96.14	89.07	96.21	67.70
	gold seg	–	–	96.31	96.67	89.75	96.29	78.08
	gold seg+mor	–	–	–	–	–	98.29	82.01
Greek-GDT	raw text	99.86	90.11	97.62	97.52	94.18	97.26	88.21
	gold seg	–	–	97.79	97.72	94.41	97.38	89.09
	gold seg+mor	–	–	–	–	–	98.17	90.17
Hebrew-HTB	raw text	99.98	100.00	82.70	82.71	81.05	82.93	64.47
	gold seg	–	–	97.23	97.28	95.59	97.19	85.76
	gold seg+mor	–	–	–	–	–	98.50	87.79
Hindi-HDTB	raw text	100.00	99.20	97.43	96.93	93.91	98.70	91.58
	gold seg	–	–	97.44	96.93	93.91	98.70	91.63
	gold seg+mor	–	–	–	–	–	98.91	94.02
Hungarian-Szeged	raw text	99.81	95.58	94.08	99.81	92.47	94.57	78.53
	gold seg	–	–	94.20	100.00	92.63	94.70	79.04
	gold seg+mor	–	–	–	–	–	97.45	83.65

Table A1. Continued

Treebank		Tokens	Sents	UPOS	XPOS	UFeats	Lemmas	LAS
Indonesian-GSD	raw text	100.00	92.00	91.93	94.52	95.59	99.68	78.34
	gold seg	-	-	91.94	94.52	95.61	99.68	78.65
	gold seg+mor	-	-	-	-	-	99.85	81.40
Irish-IDT	raw text	99.30	92.60	92.36	91.05	82.47	90.52	70.88
	gold seg	-	-	93.02	91.68	83.16	91.15	71.80
	gold seg+mor	-	-	-	-	-	94.47	74.47
Italian-ISDT	raw text	99.75	96.81	97.63	97.41	97.51	98.16	90.22
	gold seg	-	-	97.94	97.73	97.77	98.44	90.91
	gold seg+mor	-	-	-	-	-	99.35	92.70
Italian-PoSTWITA	raw text	99.73	21.80	95.71	95.38	95.76	96.63	72.22
	gold seg	-	-	96.29	95.97	96.31	97.17	81.40
	gold seg+mor	-	-	-	-	-	98.79	83.61
Japanese-GSD	raw text	90.46	95.01	88.84	90.46	90.45	89.63	74.52
	gold seg	-	-	97.84	100.00	99.98	98.78	92.43
	gold seg+mor	-	-	-	-	-	99.13	94.16
Kazakh-KTB	raw text	93.11	81.56	51.06	46.83	35.10	57.43	22.79
	gold seg	-	-	55.38	51.19	37.54	61.83	26.50
	gold seg+mor	-	-	-	-	-	95.66	33.70
Korean-GSD	raw text	99.81	90.49	96.09	90.53	99.59	93.94	83.46
	gold seg	-	-	96.35	90.76	99.79	94.11	84.68
	gold seg+mor	-	-	-	-	-	97.94	86.69
Korean-Kaist	raw text	100.00	100.00	95.61	87.15	100.00	94.39	86.77
	gold seg	-	-	95.61	87.15	100.00	94.39	86.77
	gold seg+mor	-	-	-	-	-	98.85	89.91
Kurmanji-MG	raw text	94.33	69.14	55.42	51.87	42.01	64.83	23.44
	gold seg	-	-	57.28	52.78	43.51	67.94	25.19
	gold seg+mor	-	-	-	-	-	91.72	28.88
Latin-ITTB	raw text	99.94	82.49	98.32	94.47	95.35	98.67	86.53
	gold seg	-	-	98.38	94.56	95.44	98.70	88.97
	gold seg+mor	-	-	-	-	-	99.64	91.04
Latin-PROIEL	raw text	99.99	35.16	96.61	96.81	90.94	97.20	71.47
	gold seg	-	-	96.87	97.05	91.51	97.27	80.54
	gold seg+mor	-	-	-	-	-	98.97	83.73

**Table A1.** Continued

Treebank		Tokens	Sents	UPOS	XPOS	UFeats	Lemmas	LAS
Latin-Perseus	raw text	100.00	98.35	90.52	75.01	79.18	85.27	62.27
	gold seg	-	-	90.49	75.00	79.15	85.26	62.42
	gold seg+mor	-	-	-	-	-	91.67	69.58
Latvian-LVTB	raw text	99.40	98.34	94.99	86.46	91.08	93.95	80.83
	gold seg	-	-	95.47	86.88	91.57	94.45	81.67
	gold seg+mor	-	-	-	-	-	98.53	85.30
North Sami-Giella	raw text	99.84	98.33	91.37	92.94	88.11	89.70	69.60
	gold seg	-	-	91.49	93.17	88.33	89.83	69.89
	gold seg+mor	-	-	-	-	-	94.71	80.11
Norwegian-Bokmaal	raw text	99.78	95.79	97.33	99.78	96.25	97.97	89.52
	gold seg	-	-	97.53	100.00	96.45	98.18	90.27
	gold seg+mor	-	-	-	-	-	99.62	93.12
Norwegian-Nynorsk	raw text	99.93	92.08	97.18	99.93	96.25	97.72	89.46
	gold seg	-	-	97.30	100.00	96.38	97.81	90.31
	gold seg+mor	-	-	-	-	-	99.41	92.82
Norwegian-NynorskLIA	raw text	99.99	99.86	89.56	99.99	88.84	94.51	57.45
	gold seg	-	-	89.63	100.00	88.71	94.53	57.37
	gold seg+mor	-	-	-	-	-	98.50	65.13
Old Church Slavonic-PROIEL	raw text	100.00	37.28	96.09	96.20	89.39	95.14	73.36
	gold seg	-	-	96.35	96.53	89.76	95.27	83.43
	gold seg+mor	-	-	-	-	-	97.29	87.49
Persian-Seraji	raw text	100.00	98.74	97.02	97.05	97.12	96.77	85.98
	gold seg	-	-	97.31	97.34	97.41	97.03	86.53
	gold seg+mor	-	-	-	-	-	97.29	89.62
Polish-LFG	raw text	99.86	99.74	98.26	93.54	94.57	97.66	94.51
	gold seg	-	-	98.39	93.65	94.69	97.77	94.89
	gold seg+mor	-	-	-	-	-	99.47	96.64
Polish-SZ	raw text	99.99	99.00	97.85	92.03	92.13	97.08	91.15
	gold seg	-	-	97.99	92.20	92.30	97.21	91.76
	gold seg+mor	-	-	-	-	-	98.86	94.40
Portuguese-Bosque	raw text	99.71	88.79	96.07	99.59	95.73	97.58	87.42
	gold seg	-	-	96.43	100.00	96.11	98.02	88.56
	gold seg+mor	-	-	-	-	-	98.70	89.42

Table A1. Continued

Treebank		Tokens	Sents	UPOS	XPOS	UFeats	Lemmas	LAS
Romanian-RRT	raw text	99.67	93.72	97.53	97.04	97.20	98.23	86.04
	gold seg	-	-	97.86	97.35	97.51	98.54	86.87
	gold seg+mor	-	-	-	-	-	99.63	87.53
Russian-SynTagRus	raw text	99.60	98.01	97.98	99.60	96.46	98.15	91.66
	gold seg	-	-	98.37	100.00	96.85	98.53	92.51
	gold seg+mor	-	-	-	-	-	99.60	93.63
Russian-Taiga	raw text	98.14	87.38	91.88	98.12	82.23	89.32	64.15
	gold seg	-	-	93.33	99.98	83.86	90.80	66.67
	gold seg+mor	-	-	-	-	-	96.40	69.79
Serbian-SET	raw text	99.97	92.02	97.97	99.97	93.95	97.17	88.60
	gold seg	-	-	97.99	100.00	94.00	97.20	89.11
	gold seg+mor	-	-	-	-	-	98.68	90.37
Slovak-SNK	raw text	100.00	84.26	95.69	84.30	89.94	96.35	86.61
	gold seg	-	-	95.69	84.44	90.06	96.35	88.43
	gold seg+mor	-	-	-	-	-	98.30	90.72
Slovenian-SSJ	raw text	98.29	76.61	96.49	92.34	92.82	96.49	86.78
	gold seg	-	-	98.15	94.21	94.73	98.21	91.63
	gold seg+mor	-	-	-	-	-	99.49	94.22
Slovenian-SST	raw text	100.00	22.90	93.87	85.43	85.39	94.90	53.97
	gold seg	-	-	94.35	85.45	85.54	95.04	66.26
	gold seg+mor	-	-	-	-	-	98.89	72.45
Spanish-AnCora	raw text	99.97	98.26	97.80	97.76	97.34	98.48	89.62
	gold seg	-	-	97.85	97.81	97.39	98.52	89.87
	gold seg+mor	-	-	-	-	-	99.71	91.78
Swedish-LinES	raw text	99.96	85.25	96.63	94.58	89.55	97.29	81.16
	gold seg	-	-	96.64	94.63	89.63	97.34	82.23
	gold seg+mor	-	-	-	-	-	98.41	84.06
Swedish-PUD	raw text	98.41	94.47	93.58	91.88	77.99	87.47	79.15
	gold seg	-	-	94.31	92.48	78.80	88.86	80.42
	gold seg+mor	-	-	-	-	-	90.42	82.90
Swedish-Talbanken	raw text	99.78	93.17	97.47	96.41	96.63	97.98	85.87
	gold seg	-	-	97.64	96.55	96.77	98.15	86.74
	gold seg+mor	-	-	-	-	-	98.70	89.10

Table A1. Continued

Treebank		Tokens	Sents	UPOS	XPOS	UFeats	Lemmas	LAS
Turkish-IMST	raw text	99.86	97.09	94.32	93.27	91.05	95.16	64.70
	gold seg	-	-	96.19	95.02	92.83	97.03	67.84
	gold seg+mor	-	-	-	-	-	99.42	69.95
Ukrainian-IU	raw text	99.67	95.04	97.04	90.71	90.83	96.62	84.43
	gold seg	-	-	97.41	90.96	91.09	96.93	85.24
	gold seg+mor	-	-	-	-	-	99.16	88.00
Upper Sorbian-UFAL	raw text	98.60	74.51	59.51	98.60	39.66	54.80	24.90
	gold seg	-	-	60.42	100.00	40.58	55.26	26.13
	gold seg+mor	-	-	-	-	-	73.74	34.51
Urdu-UDTB	raw text	100.00	98.60	94.54	92.74	83.90	97.43	82.15
	gold seg	-	-	94.54	92.73	83.91	97.43	82.20
	gold seg+mor	-	-	-	-	-	97.99	86.85
Uyghur-UDT	raw text	99.22	81.61	89.15	91.56	87.45	94.15	62.92
	gold seg	-	-	89.96	92.24	88.16	94.95	65.07
	gold seg+mor	-	-	-	-	-	99.01	68.44
Vietnamese-VTB	raw text	84.26	92.87	76.50	73.79	83.93	84.26	42.87
	gold seg	-	-	89.15	85.36	99.53	99.98	59.88
	gold seg+mor	-	-	-	-	-	99.98	69.38

**Cite this article:** Kanerva J, Ginter F and Salakoski T (2021). Universal Lemmatizer: A sequence-to-sequence model for lemmatizing Universal Dependencies treebanks. *Natural Language Engineering* 27, 545–574. <https://doi.org/10.1017/S1351324920000224>





**Jenna Kanerva & Filip Ginter & Niko Miekka & Akseli Leino  
& Tapio Salakoski**  
**Turku Neural Parser Pipeline: An End-to-End System for the  
CoNLL 2018 Shared Task**

In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from  
Raw Text to Universal Dependencies. 2018.



# Turku Neural Parser Pipeline: An End-to-End System for the CoNLL 2018 Shared Task

Jenna Kanerva<sup>1,2</sup> Filip Ginter<sup>1</sup> Niko Miekka<sup>1</sup> Akseli Leino<sup>1</sup> Tapio Salakoski<sup>1</sup>

<sup>1</sup>Turku NLP Group, Department of Future Technologies, University of Turku, Finland

<sup>2</sup>University of Turku Graduate School (UTUGS)

firstname.lastname@utu.fi

## Abstract

In this paper we describe the TurkuNLP entry at the *CoNLL 2018 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies*. Compared to the last year, this year the shared task includes two new main metrics to measure the morphological tagging and lemmatization accuracies in addition to syntactic trees. Basing our motivation into these new metrics, we developed an end-to-end parsing pipeline especially focusing on developing a novel and state-of-the-art component for lemmatization. Our system reached the highest aggregate ranking on three main metrics out of 26 teams by achieving 1st place on metric involving lemmatization, and 2nd on both morphological tagging and parsing.

## 1 Introduction

The 2017 and 2018 CoNLL UD Shared tasks aim at an evaluation of end-to-end parsing systems on a large set of treebanks and languages. The 2017 task (Zeman et al., 2017) focused primarily on the evaluation of the syntactic trees produced by the participating systems, whereas the 2018 task (Zeman et al., 2018) adds further two metrics which also measure the accuracy of morphological tagging and lemmatization. In this paper, we present the TurkuNLP system submission to the *CoNLL 2018 UD Shared Task*. The system is an end-to-end parsing pipeline, with components for segmentation, morphological tagging, parsing, and lemmatization. The tagger and parser are based on the 2017 winning system by Dozat et al. (2017), while the lemmatizer is a novel approach utilizing the OpenNMT neural machine translation system for sequence-to-sequence learning. Our pipeline

ranked first on the evaluation metric related to lemmatization, and second on the metrics related to tagging and parsing.

## 2 Task overview

*CoNLL 2018 UD Shared Task* is a follow-up to the 2017 shared task of developing systems predicting syntactic dependencies on raw texts across a number of typologically different languages. In addition to the 82 UD treebanks for 57 languages, which formed the primary training data, the participating teams were allowed to use also additional resources such as Wikipedia dumps<sup>1</sup>, raw web crawl data and word embeddings (Ginter et al., 2017), morphological transducers provided by Apertium<sup>2</sup> and Giellatekno<sup>3</sup>, and the OPUS parallel corpus collection (Tiedemann, 2012). In addition to the 2017 primary metric (LAS), the systems were additionally evaluated also on metrics which include lemmatization and morphology prediction. In brief, the three primary metrics of the task are as follows (see Zeman et al. (2018) for detailed definitions):

**LAS** The proportion of words which have the correct head word with the correct dependency relation.

**MLAS** Similar to LAS, with the additional requirement that a subset of the morphology features is correctly predicted and the functional dependents of the word are correctly attached. MLAS is only calculated on content-bearing words, and strives to level the field w.r.t. morphological richness of languages.

<sup>1</sup><https://dumps.wikimedia.org>

<sup>2</sup><https://svn.code.sf.net/p/apertium/svn/languages>

<sup>3</sup><https://victorio.uit.no/langtech/trunk/langs>

**BLEX** The proportion of head-dependent content word pairs whose dependency relation and both lemmas are correct.

### 3 System overview and rationale

The design of the pipeline was dictated by the tight schedule and the limited manpower we were able to invest into its development. Our overall objective was to develop an easy-to-use parsing pipeline which carries out all the four tasks of segmentation, morphological tagging, parsing, and lemmatization, resulting in an end-to-end full parsing pipeline reusable in downstream applications. We also strove for the pipeline to perform well on all four tasks and all groups of treebanks, ranging from the large treebanks to the highly under-resourced ones. With this in mind, we decided to rely on openly available components when the acceptable performance is already met, and create our own components for those tasks we see clear room for improvement.

Therefore, for segmentation, tagging and parsing we leaned as much as possible on well-known components trained in the standard manner, and deviated from these only when necessary. Our approach to lemmatization, on the other hand, is original and previously unpublished. In summary, we rely for most but not all languages on the tokenization and sentence splitting provided by the UDPipe baseline (Straka et al., 2016). Tagging and parsing is carried out using the parser of Dozat et al. (2017), the winning entry of the 2017 shared task. Using a simple data manipulation technique, we also obtain the morphological feature predictions from the same tagger which was originally used to produce only universal part-of-speech (UPOS) and language-specific part-of-speech (XPOS) predictions. Finally, the lemmatization is carried out using the OpenNMT neural machine translation toolkit (Klein et al., 2017), casting lemmatization as a machine translation problem. All these components are wrapped into one parsing pipeline, making it possible to run all four steps with one simple command and gain state-of-the-art or very close to state-of-the-art results for each step. In the following, we describe each of these four steps in more detail, while more detailed description of the pipeline itself is given in Section 6.

#### 3.1 Tokenization and sentence splitting

For all but three languages, we rely on the UDPipe baseline runs provided by the shared task organizers. The three languages where we decided to deviate from the baseline are Thai, Breton and Faroese. Especially for Thai we suspected the UDPipe baseline, trained without ever seeing a single character of the Thai alphabet, would perform poorly. For Breton, we were unsure about the way in which the baseline system tokenizes words with apostrophes like *arc'hant* (money), and without deeper knowledge of Breton language decided that it is better to explicitly keep all words with apostrophes unsegmented. We therefore developed a regular-expression based sentence splitter and tokenizer — admittedly under a very rushed schedule — which splits sentences and tokens on a handful of punctuation characters. While, after the fact, we can see that the UDPipe baseline performed well at 92.3%, our solution outperformed it by two percentage points, validating our choice. For Thai, we developed our own training corpus using machine translation (described later in the paper in Section 4.3), and trained UDPipe on this corpus, gaining a segmentation model at the same time. Indeed, the UDPipe baseline only reached 8.5% accuracy while our tokenizer performed at the much higher 43.2% (still far below the 70% achieved by the Uppsala team). Similarly, for Faroese we built training data by pooling the Danish-DDT, Swedish-Talbanken, and the three available Norwegian treebanks (Bokmaal, Nynorsk, NynorskLIA), and subsequently trained the UDPipe tokenizer on this data. After the fact, we can see that essentially all systems performed in the 99–100% range on Faroese, and we could have relied on the UDPipe baseline.

On a side note, we did develop our own method for tokenization and sentence splitting but in the end, unsure about its stability and performance on small treebanks, we decided to “play it safe” and not include it in the final system. However, the newly developed tokenizer is part of our open-source pipeline release and trainable on new data.

#### 3.2 Pre-trained embeddings

Where available, we used the pre-trained embeddings from the 2017 shared task (Ginter et al., 2017). Embeddings for Afrikaans, Breton, Buryat, Faroese, Gothic, Upper Sorbian, Armenian, Kurdish, Northern Sami, Serbian and Thai were ob-

tained from the embeddings published by Facebook<sup>4</sup> trained using the fastText method (Bojanowski et al., 2016), and finally for Old French (Old French-SRCMF) we took the embeddings trained using word2vec (Mikolov et al., 2013) on the treebank train section by the organizers in their baseline UDPipe model release. We did not pre-train any embeddings ourselves.

### 3.3 UPOS tagging

UPOS tagging for all languages is carried out using the system of Dozat et al. (2017) trained out-of-the-box with the default set of parameters from the CoNLL-17 shared task. The part-of-speech tagger is a time-distributed affine classifier over tokens in a sentence, where tokens are first embedded with a word encoder which sums together a learned token embedding, a pre-trained token embedding and a token embedding encoded from the sequence of its characters using unidirectional LSTM. After that bidirectional LSTM reads the sequence of embedded tokens in a sentence to create a context-aware token representations. These token representations are then transformed with ReLU layers separately for each affine tag classification layers (namely UPOS and XPOS). These two classification layers are trained jointly by summing their cross-entropy losses. For more detailed description, see Dozat and Manning (2016) and Dozat et al. (2017).

### 3.4 XPOS and FEATS tagging

As the tagger of Dozat et al. predicts the XPOS field, we used a simple trick of concatenating the FEATS field into XPOS, therefore manipulating the tagger into predicting the XPOS and morphological features as one long string. For example the original XPOS field value N and FEATS field value `Case=Nom|Number=Sing` in Finnish-TDT treebank gets concatenated into `XPOS=N|Case=Nom|Number=Sing` and this full string is predicted as one class by the tagger. After tagging and parsing, these values are again splitted into correct columns. This is a (embarrassingly) simple approach which leads to surprisingly good results, as our system ranks 3rd in morphological features with accuracy of 86.7% over all treebanks, 0.9pp below the Uppsala team which ranked 1st on this subtask.

<sup>4</sup><https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md>

We, in fact, did at first develop a comparatively complex morphological feature prediction component which outperformed the state-of-the-art on the 2017 shared task, but later we discovered that the simple technique described above somewhat surprisingly gives notably better results. We expected that the complex morphology of many languages leads to a large number of very rare morphological feature strings, a setting unsuitable for casting the problem as a single multi-class prediction task. Consequently, our original attempt at morphological tagging predicted value for each morphological category separately from a shared representation layer, rather than predicting the full feature string at once. To shed some light on the complexity of the problem in terms of the number of classes, and understand why a multi-class setting works well, we list in Table 1 the number of unique morphological feature strings needed to cover 80%, 90%, 95%, and 100% of the running words in the training data for each language. The number of unique feature combinations varies from 15 (Japanese-GSD, Vietnamese-VTB) to 2629 (Czech-PDT), and for languages with high number of unique combinations, we can clearly see that there is a large leap from covering 95% of running words to covering full 100%. For example in Czech-PDT, only 349 out of the 2629 feature combinations are needed to cover 95% of running words, and the rest 2280 (of which 588 are singletons) together accounts only 5% of running words. Based on these numbers our conclusions are that a focus on predicting the rare feature combinations correctly does not affect the accuracy much, and learning a reasonable number of common feature combinations well seems to be a good strategy in the end.

Interestingly, on our preliminary experiments with Finnish, we found that concatenating FEATS into XPOS improved also LAS by more than 0.5pp, since the parser takes the XPOS field as a feature and benefits from the additional morphological information present. To investigate this more closely and test whether the same improvement can be seen on other languages as well, we carry out an experiment where we train the tagger and parser without morphological information for Finnish and six more arbitrarily chosen treebanks. This new experiment then follows the original training setting used by the Stanford team on their CoNLL-17 submission, and by comparing this to

	80%	90%	95%	100%		80%	90%	95%	100%
Czech-PDT	96	194	349	2629	Arabic-PADT	22	35	53	322
Finnish-TDT	79	188	349	2052	Spanish-AnCora	28	48	71	295
Finnish-FTB	72	174	333	1762	Italian-ISDT	22	35	55	281
Czech-CAC	81	160	285	1745	Catalan-AnCora	28	47	68	267
Czech-FicTree	73	161	287	1464	French-GSD	19	31	46	225
Slovak-SNK	79	163	283	1199	Italian-PoSTWITA	23	39	56	224
Ukrainian-IU	91	186	322	1197	Galician-TreeGal	23	41	66	222
Polish-LFG	84	170	281	1171	Uyghur-UDT	21	40	63	214
Slovenian-SSJ	73	141	254	1101	Swedish-Talbanken	26	43	61	203
Croatian-SET	63	125	212	1099	Norwegian-Bokmaal	26	39	57	203
Latin-PROIEL	121	214	323	1031	French-Sequoia	25	43	62	200
Ancient Greek-PROIEL	114	203	308	1027	Indonesian-GSD	12	20	31	192
Urdu-UDTB	30	61	124	1001	Norwegian-Nynorsk	26	41	53	184
Polish-SZ	80	157	267	991	Swedish-LinES	25	43	61	173
Latin-ITTB	58	136	226	985	Persian-Seraji	11	19	31	162
Turkish-IMST	54	139	262	972	Danish-DDT	24	38	53	157
Hindi-HDTB	38	76	127	939	Armenian-ArmTDP	51	85	117	157
Estonian-EDT	43	89	151	918	English-EWT	19	32	45	150
German-GSD	58	96	141	909	Upper Sorbian-UFAL	48	88	111	134
Basque-BDT	51	100	169	884	English-LinES	18	29	43	104
Old Church Slavonic-PROIEL	78	168	276	859	English-GUM	16	27	40	104
Latvian-LVTB	57	119	218	828	Kazakh-KTB	29	49	71	98
Ancient Greek-Perseus	59	107	169	774	Norwegian-NynorskLIA	22	34	46	96
Russian-SynTagRus	67	124	176	734	Dutch-Alpino	16	24	31	63
Slovenian-SST	73	146	233	645	Afrikaans-AfriBooms	14	22	28	61
Gothic-PROIEL	75	138	214	623	Dutch-LassySmall	13	19	26	59
Hungarian-Szeged	40	90	166	581	Kurmanji-MG	24	35	46	58
Serbian-SET	48	85	131	539	Old French-SRCMF	11	15	19	57
Hebrew-HTB	19	45	85	521	Buryat-BDT	17	26	34	41
Romanian-RRT	34	58	97	451	Chinese-GSD	7	10	13	31
Bulgarian-BTB	33	63	107	432	Galician-CTG	7	9	11	27
Latin-Perseus	58	100	144	418	Korean-GSD	4	4	6	19
Portuguese-Bosque	20	35	60	396	Korean-Kaist	6	8	10	17
Russian-Taiga	66	126	182	376	French-Spoken	8	10	12	16
North Sami-Giella	39	78	127	369	Vietnamese-VTB	6	8	10	15
Irish-IDT	47	81	125	360	Japanese-GSD	5	7	9	15
Greek-GDT	57	90	123	348					

Table 1: The number of unique UPOS+morphological feature combinations needed to cover 80%, 90%, 95% and 100% of the running words in each treebank.

our main runs we can directly evaluate the effect of predicting additional morphological information. Three of the treebanks used in this experiment (Arabic-PADT, Czech-PDT and Swedish-Talbanken) seem to originally encode the full (or at least almost full) morphological information in the XPOS field in a language-specific manner (e.g. AAFS1----2A---- in Czech), whereas four treebanks seem to include only part-of-speech like information or nothing at all in the XPOS field (Estonian-EDT, Finnish-TDT, Irish-IDT and Russian-SynTagRus).

The results of this experiment are shown in Table 2. Four treebanks above the dashed line, those originally including only part-of-speech like information in the XPOS field, shows clear positive im-

provement in terms of LAS when the parser is able to see also morphological tags predicted together with the language-specific XPOS. The parser seeing the morphological tags (LAS<sub>m</sub> column) shows improvements approx. from +0.3 to +0.9 for these four treebanks compared to the parser without morphological tags (LAS column). Three treebanks below the dashed line, those already including language-specific morphological information in the XPOS field, quite naturally does not benefit from additional morphology and shows mildly negative results in terms of LAS. However the difference in treebanks showing negative results is substantially smaller compared to those having positive effect (negative differences stay between -0.0 to -0.2), therefore based on these seven tree-

Treebank	LAS	LAS <sub>m</sub>		UPOS	UPOS <sub>m</sub>		XPOS	XPOS <sub>m</sub>	
Estonian-EDT	83.40	<b>84.15</b>	(+0.75)	96.32	<b>96.45</b>	(+0.13)	97.81	<b>97.87</b>	(+0.06)
Finnish-TDT	85.74	<b>86.60</b>	(+0.86)	96.45	<b>96.66</b>	(+0.21)	97.48	<b>97.63</b>	(+0.15)
Irish-IDT	70.01	<b>70.88</b>	(+0.87)	91.87	<b>92.36</b>	(+0.49)	91.01	<b>91.05</b>	(+0.04)
Russian-SynT.	91.40	<b>91.72</b>	(+0.32)	<b>98.11</b>	98.03	(-0.08)	—	—	
Arabic-PADT	<b>72.67</b>	72.45	(-0.22)	90.39	<b>90.48</b>	(+0.19)	87.36	<b>87.39</b>	(+0.03)
Czech-PDT	<b>90.62</b>	90.57	(-0.05)	<b>98.76</b>	98.74	(-0.02)	<b>95.66</b>	95.44	(-0.22)
Swedish-Talb.	<b>85.87</b>	85.83	(-0.04)	97.40	<b>97.47</b>	(+0.07)	96.36	<b>96.41</b>	(+0.05)

Table 2: LAS, UPOS and XPOS scores for seven parsers trained with and without tagger predicting the additional morphological information. *m* after the score name stands for including the morphological information during training, i.e. the official result for our system. Note that when evaluating XPOS, the morphological information is already extracted from that field so the evaluation only includes prediction of original XPOS-tags, not morphological features.

banks the overall impact stays on positive side. Note that during parsing the parser only sees predicted morphological features, so this experiment confirms that predicting more complex information on lower-level can improve the parser.

Because of the fact that many treebanks include more than plain part-of-speech information in the language-specific XPOS field, likely more natural place for the morphological features would be the universal part-of-speech field UPOS which is guaranteed to include only universal part-of-speech information. However, with the limited time we had during the shared task period, we had no time to test whether adding morphological features harms the prediction of original part-of-speech tag, and we decided to use XPOS field as we thought it’s least important of these two. Based on the results in the XPOS column of Table 2, we however see that additional information does not generally seem to harm the prediction of the original language-specific part-of-speech tags and hints towards the conclusion that likely the UPOS field could have been used with comparable performance.

### 3.5 Syntactic parsing

Syntactic parsing for all languages is carried out using the system of Dozat et al. trained out-of-the-box with the default set of parameters from the CoNLL-17 shared task. The parser architecture is quite similar as used in the tagger. Tokens are first embedded with a word encoder which sums together a learned token embedding, a pre-trained token embedding and a token embedding encoded from the sequence of its characters using unidirectional LSTM. These embedded tokens

are yet concatenated together with corresponding part-of-speech embeddings. After that bidirectional LSTM reads the sequence of embedded tokens in a sentence to create a context-aware token representations. These token representations are then transformed with four different ReLU layers separately for two different biaffine classifiers to score possible relations (HEAD) and their dependency types (DEPREL), and best predictions are later decoded to form a tree. These relation and type classifiers are again trained jointly by summing their cross-entropy losses. For more detailed description, see Dozat and Manning (2016) and Dozat et al. (2017).

### 3.6 Lemmatization

While in many real word industry applications especially for inflective languages the lemmatizer is actually the most needed component of the parsing pipeline, yet it’s performance has been undesirable weak in previous state-of-the-art parsing pipelines for many inflectionally complex languages. For this reason we develop a novel and previously unpublished component for lemmatization.

We represent lemmatization as a sequence-to-sequence translation problem, where the input is a word represented as a sequence of characters concatenated with a sequence of its part-of-speech and morphological tags, while the desired output is the corresponding lemma represented as a sequence of characters. Therefore we are training the system to translate the word form characters + morphological tags into the lemma characters, where each word is processed independently from it’s sentence context. For example, input and output sequences for the English word *circles* as a

noun are:

```
INPUT: c i r c l e s UPOS=NOUN  
      XPOS=NNS Number=Plur
```

```
OUTPUT: c i r c l e
```

As our approach can be seen similar to general machine translation problem, we are able to use any openly available machine translation toolkit and translation model implementations. Our current implementation is based on the Python version of the OpenNMT: Open-Source Toolkit for Neural Machine Translation (Klein et al., 2017). We use a deep attentional encoder-decoder network with 2 layered bidirectional LSTM encoder for reading the sequence of input characters + morphological tags and producing a sequence of encoded vectors. Our decoder is a 2 layered unidirectional LSTM with input feeding attention for generating the sequence of output characters based on the encoded representations. In input feeding attention (Luong et al., 2015) the previous attention weights are given as input in the next time step to inform the model about past alignment decisions and prevent the model to repeat the same output multiple times. We use beam search with beam size 5 during decoding.

As the lemmatizer does not see the actual sentence where a word appears, morphological tags are used in the input sequence to inform the system about the word’s morpho-syntactic context. The tagger is naturally able to see the full sentence context and in most cases it should produce enough information for the lemmatizer to give it a possibility to lemmatize ambiguous words correctly based on the current context. During test time we run the lemmatizer as a final step in the parsing pipeline, i.e. after tagger and parser, so the lemmatizer runs on top of the predicted part-of-speech and morphological features. Adding the lemmatizer only after the tagger and parser (and not before like done in many pipelines) does not cause any degradation for the current pipeline as the tagger and parser by Dozat et al. (2017) do not use lemmas as features.

This method is inspired by the top systems from the CoNLL-SIGMORPHON 2017 Shared Task of Universal Morphological Reinflection (Cotterell et al., 2017), where the participants used encoder-decoder networks to generate inflected words from the lemma and given morphological tags (Kann and Schütze, 2017; Bergmanis et al., 2017). While

the SIGMORPHON 2017 Shared Task was based on gold standard input features, to our knowledge we are the first ones to use similar techniques on reversed problem settings and to incorporate such lemmatizer into the full parsing pipeline to run on top of predicted morphological features.

## 4 Near-zero resource languages

There are nine very low resource languages: Breton, Faroese, Naija and Thai with no training data, and Armenian, Buryat, Kazakh, Kurmanji and Upper Sorbian with only a tiny training dataset. For the latter five treebanks with tiny training sample, we trained the tagger and parser in the standard manner, despite the tiny training set size. However, for four of these five languages (Armenian, Buryat, Kazakh and Kurmanji) we used Apertium morphological transducers (Tyers et al., 2010) to artificially extend the lemmatizer training data by including new words from the transducer not present in the original training data (methods are similar to those used with Breton and Faroese, for details see Section 4.1). Naija is parsed using the English-EWT models without any extra processing as it strongly resembles English language and at the same time lacks all resources. Breton, Faroese and Thai were each treated in a different manner described below.

### 4.1 Breton

Our approach to Breton was to first build a Breton POS and morphological tagger, and subsequently apply a delexicalized parser. To build the tagger, we selected 5000 random sentences from the Breton Wikipedia text dump and for each word looked up all applicable morphological analyzes in the Breton Apertium transducer converted into UD using a simple language-agnostic mapping from Apertium tags to UD tags. For words unknown to the transducer (59% of unique words), we assign all possible UPOS+FEATS strings produced by the transducer on the words it recognizes in the data. Then we decode the most likely sequence of morphological readings using a delexicalized 3-gram language model trained on the UPOS+FEATS sequences of English-EWT and French-GSD training data. Here we used the `lazy` decoder program<sup>5</sup> which is based on the KenLM language model estimation and querying system (Heafield, 2011). This procedure re-

<sup>5</sup><https://github.com/kpu/lazy>

sults in 5000 sentences (96,304 tokens) of morphologically tagged Breton, which can be used to train the tagger in the usual manner. The syntactic parser was trained as delexicalized (FORM field replaced with underscore) on the English-EWT and French-GSD treebanks. The accuracy of UPOS and FEATS was 72% (3rd rank) and 56.6% (2nd rank) and LAS ranked 3rd with 31.8%. These ranks show our approach as competitive in the shared task, nevertheless the Uppsala team achieved some 14pp higher accuracies of UPOS and FEATS, clearly using a considerably better approach.

The Breton lemmatizer was trained using the same training data as used for the tagger, where for words recognized by the transducer the part-of-speech tag and morphological features are converted into UD with the language-agnostic mapping, and lemmas are used directly. Unknown words for transducer (i.e. those for which we are not able to get any lemma analysis) are simply skipped from the lemmatizer training. As the lemmatizer sees each word separately, skipping words and breaking the sentence context does not cause any problems. With this approach we achieved the 1st rank and accuracy of 77.6%, which is over 20pp better than the second best team.

To estimate the quality of our automatically produced training data for Breton tagging and lemmatization, we repeat the same procedure with the Breton test data<sup>6</sup>, i.e. we use the combination of morphological transducer and language model as a direct tagger leaving out the part of training an actual tagger with the produced data as done in our original method. When evaluating these produced analyses against the gold standard, we get a direct measure of quality for this method. We measure three different scores: 1) Oracle full match of transducer readings converted to UD, where we measure how many tokens can receive a correct combination of UPOS and all morphological tags when taking into account all possible readings given by the transducer. For unknown words we include all combinations known from the transducer. This setting measures the best full match number achievable by the language model if it would predict everything perfectly. 2) Language model full match, i.e. how many tokens received a fully correct analysis when lan-

<sup>6</sup>Using development data in these experiments would be more desirable, but unfortunately we don't have any Breton development data available.

guage model was used to pick one of the possible analyses. 3) Random choice full match, i.e. how many tokens received a fully correct analysis when one of the possible analyses was picked randomly. On Breton test set our oracle full match is 55.5%, language model full match 51.0% and random full match 46.2%. We can see that using a language model to pick analyses shifts the performance more closer to oracle full match than random full match, showing somewhat positive results for the language model decoding. Unfortunately when we tried to replicate the same experiment for other low-resource languages, we did not see the same positive signal. However, the biggest weakness of this method seems to be in the oracle full match which is only 55.5%. This means that the correct analysis cannot be found from the converted transducer output for almost half of the tokens. A probable reason for this is the simple language-agnostic mapping from Apertium tags to UD tags which is originally developed for the lemmatizer training and strove for high precision rather than high recall. Our development hypothesis was that missing a tag in lemmatizer's input likely does not tremendously harm the lemmatizer, so when developing the mapping we rather left some tags out than caused a potential erroneous conversion. However, when the same mapping is used here, missing one common tag (for example `VerbForm=Fin`) can cause great losses in full match evaluation.

## 4.2 Faroese

For Faroese the starting situation was similar to Breton but as the coverage of the Faroese Apertium transducer was weak, we decided to take another approach. This is because we feared that the decoder input would have too many gaps to fill in and therefore the quality of produced data would decrease. For that reason the Faroese tagger and parser was trained in the usual manner using pooled training sets of related Nordic languages: Danish-DDT, Swedish-Talbanken, and the three available Norwegian treebanks (Bokmaal, Nynorsk, NynorskLIA). The pre-trained embeddings were Faroese from the Facebook's embeddings dataset, filtered to only contain words which Faroese has in common with one of the languages used in training. However, the Faroese lemmatizer is trained directly from the transducer output by analyzing vocabulary extracted from the



Faroese Wikipedia and turning Apertium analyses into UD using the same tag mapping table as in the Breton. On UPOS tagging our system ranks only 10th, whereas on both morphological feature prediction and lemmatization, we rank 1st.

### 4.3 Thai

As there is no training data and no Apertium morphological transducer for Thai, we machine translated the English-EWT treebank word-for-word into Thai, and used the result as training data for the Thai segmenter, tagger and parser. Here we utilized the Marian neural machine translation framework (Junczys-Dowmunt et al., 2018) trained on the 6.1 million parallel Thai-English sentences in OPUS (Tiedemann, 2012). Since we did not have access to a Thai tokenizer and Thai language does not separate words with spaces, we forced the NMT system into character-level mode by inserting a space between all characters in a sentence (both on the source and the target side) and again removing those after translation. After training the translation system, the English-EWT treebank is translated one word at a time, creating a token and sentence segmented Thai version of the treebank. Later all occurrences of English dots and commas were replaced with whitespaces in the raw input text (and accordingly absence of *SpaceAfter=No* tags in CoNLL-U) as Thai uses whitespace rather than punctuation as pause character, and rest of the words were merged together in raw text by including *SpaceAfter=No* feature for each word not followed by dot or comma. This word-by-word translation and Thai word merging technique gives us the possibility to train a somewhat decent sentence and word segmenter without any training data for a language which does not use whitespaces to separate words or even sentences. Furthermore, all *the* words were removed as they have no Thai counterpart, lemmas were dropped, all matching morphological features between English and Thai were copied, HEAD indices were updated because of removing before mentioned tokens, non-existent dependency relations in Thai were mapped to similar existent ones, and finally enhanced dependency graphs were dropped. The tagger and parser were then trained normally using this training data. Training a lemmatizer is not needed as the Thai treebank does not include lemma annotation.

Our Thai segmentation achieves 1st rank and

accuracy of 12.4% on sentence segmentation and 5th rank and accuracy of 43.2% on tokenization. On UPOS prediction we have accuracy of 27.6% and 4th rank, and our LAS is 6.9% and we rank 2nd, while the best team on Thai LAS, CUNI x-ling, achieves 13.7%. English is not a particularly natural choice for the source language of a Thai parser, with Chinese likely being a better candidate. We still chose English because we were unable to train a good Chinese-Thai MT system on the data provided in OPUS and the time pressure of the shared task prevented us from exploring other possibilities. Clearly, bad segmentation scores significantly affect other scores as well, and when the parser and tagger are evaluated on top of gold segmentation, our UPOS accuracy is 49.8% and LAS 20.4%. These numbers are clearly better than with predicted segmentation but still far off from typical supervised numbers.

## 5 Results

The overall results of our system are summarized in Table 3, showing the absolute performance, rank, and difference to the best system / next best system for all metrics on several treebank groups — big, small, low-resource and parallel UD (PUD). With respect to the three main metrics of the task, we ranked 2nd on LAS, 2nd on MLAS and 1st on BLEX, and received the highest aggregate ranking out of 26 teams, of which 21 submitted non-zero runs for all treebanks. For LAS, our high rank is clearly due to balanced performance across all treebank groups, as our ranks in the individual groups are 3rd, 6th, 4th and 6th, still giving a 2nd overall rank. A similar pattern can also be observed for MLAS. Our 1st overall rank on the BLEX metric is undoubtedly due to the good performance in lemmatization, on which our system achieves the 1st rank overall as well as in all corpus groups except the low-resourced languages. Altogether, it can be seen in the results table that the two main strengths of the system is 1) lemmatization and 2) tagging of small treebanks, and on any metric, the system ranks between 1st and 5th place across all corpora (*all* column in Table 3).

## 6 Software release

The full parsing pipeline is available at <https://turkunlp.github.com/Turku-neural-parser-pipeline>,

	<i>All</i>	<i>Big</i>	<i>PUD</i>	<i>Small</i>	<i>Low</i>
LAS	73.28 (-2.56 / 2)	81.85 (-2.52 / 3)	71.78 (-2.42 / 6)	64.48 (-5.05 / 4)	22.91 (-4.98 / 6)
MLAS	60.99 (-0.26 / 2)	71.27 (-1.40 / 3)	57.54 (-1.21 / 5)	47.63 (-1.61 / 2)	3.59 (-2.54 / 5)
BLEX	<b>66.09 (+0.76 / 1)</b>	<b>75.83 (+0.37 / 1)</b>	<b>63.25 (+0.91 / 1)</b>	53.54 (-1.35 / 2)	11.40 (-2.58 / 2)
UAS	77.97 (-2.54 / 4)	85.32 (-2.29 / 5)	75.58 (-2.84 / 6)	71.50 (-4.44 / 5)	34.51 (-4.72 / 6)
CLAS	69.40 (-2.96 / 2)	78.26 (-3.03 / 4)	67.65 (-2.21 / 5)	59.28 (-5.57 / 4)	18.15 (-4.03 / 6)
UPOS tagging	89.81 (-1.10 / 4)	95.41 (-0.82 / 6)	85.59 (-1.92 / 9)	91.93 (-0.91 / 3)	52.53 (-8.54 / 4)
XPOS tagging	86.17 (-0.50 / 3)	94.47 (-0.69 / 4)	55.68 (-0.30 / 2)	<b>90.51 (+0.50 / 1)</b>	43.43 (-11.3 / 17)
Morph. features	86.70 (-0.89 / 3)	93.82 (-0.32 / 3)	85.24 (-1.81 / 5)	<b>85.63 (+0.58 / 1)</b>	40.04 (-8.91 / 4)
All morph. tags	79.83(-0.47 / 2)	91.08 (-0.42 / 3)	51.60 (-0.30 / 2)	<b>82.02 (+1.17 / 1)</b>	17.58 (-8.33 / 19)
Lemmatization	<b>91.24 (+1.92 / 1)</b>	<b>96.08 (+0.83 / 1)</b>	<b>85.76 (+0.07 / 1)</b>	<b>91.02 (+1.02 / 1)</b>	61.61 (-2.81 / 3)
Sentence segmt.	83.03 (-0.84 / 5)	86.09 (-3.43 / 7-21)	75.53 (-0.51 / 3-17)	83.33 (-0.12 / 2-20)	66.23 (-1.27 / 2)
Word segmt.	97.42 (-0.76 / 5)	98.81 (-0.40 / 8-21)	92.61 (-1.96 / 7-19)	99.43 (+0.20 / 1-19)	89.10 (-4.28 / 5)
Tokenization	97.83 (-0.59 / 4)	99.24 (-0.27 / 6-21)	92.61 (-1.96 / 7-19)	99.57 (+0.01 / 1-18)	89.85 (-3.49 / 5)

Table 3: Results in every treebank group, shown as “absolute score (difference / rank)”. For first rank, the difference to the next best system is shown, for other ranks we show the difference to the best ranking system, shared ranks are shown as a range.

together with all the trained models. We have ported the parser of Dozat et al. into Python3, and included other modifications such as the ability to parse a stream of input data without reloading the model. The pipeline has a modular structure, which allowed us to easily reconfigure the components for languages which needed a non-standard treatment. The pipeline software is documented, and we expect it to be comparatively easy to extend it with own components.

## 7 Conclusions

In this paper we presented the TurkuNLP entry at the *CoNLL 2018 UD Shared Task*. This year we focused on building an end-to-end pipeline system for segmentation, morphological tagging, syntactic parsing and lemmatization based on well-known components, and including our novel lemmatization approach. On BLEX evaluation, a metric including lemmatization and syntactic tree, we rank 1st, reflecting the state-of-the-art performance on lemmatization. On MLAS and LAS, metrics including morphological tagging and syntactic tree, and plain syntactic tree, we rank 2nd on both. All these components are wrapped into one simple parsing pipeline that carries out all four tasks with one command, and the pipeline is available for everyone together with all trained models.

## Acknowledgments

We would like to thank Tim Dozat and rest of the Stanford team for making their parser open-source, as well as Milan Straka and rest of the Prague team for making UDPipe software and

models open-source. This work was supported by Academy of Finland, Nokia Foundation and Google Digital News Innovation Fund. Computational resources were provided by CSC – IT Center for Science, Finland.

## References

- Toms Bergmanis, Katharina Kann, Hinrich Schütze, and Sharon Goldwater. 2017. Training data augmentation for low-resource morphological inflection. In *Proceedings of the CoNLL SIG-MORPHON 2017 Shared Task: Universal Morphological Reinflection*. Association for Computational Linguistics, Vancouver, pages 31–39. <http://www.aclweb.org/anthology/K17-2002>.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.
- Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. Conll-sigmorphon 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*. Association for Computational Linguistics, Vancouver, pages 1–30. <http://www.aclweb.org/anthology/K17-2001>.
- Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.
- Timothy Dozat, Peng Qi, and Christopher D Manning. 2017. Stanford’s graph-based neural dependency parser at the conll 2017 shared task. *Proceedings*

- of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies pages 20–30.
- Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 shared task - automatically annotated raw texts and word embeddings. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (UFAL), Faculty of Mathematics and Physics, Charles University. <http://hdl.handle.net/11234/1-1989>.
- Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland, United Kingdom, pages 187–197. <https://kheafield.com/papers/avenue/kenlm.pdf>.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia. <https://arxiv.org/abs/1804.00344>.
- Katharina Kann and Hinrich Schütze. 2017. The lmu system for the conll-sigmorphon 2017 shared task on universal morphological inflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Inflection*. Association for Computational Linguistics, Vancouver, pages 40–48. <http://www.aclweb.org/anthology/K17-2003>.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of the 55th annual meeting of the Association for Computational Linguistics (ACL'17)*.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1412–1421. <http://aclweb.org/anthology/D15-1166>.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.* pages 3111–3119. <http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality>.
- Milan Straka, Jan Hajič, and Jana Straková. 2016. UD-Pipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, Portoro, Slovenia.
- Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odiijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), Istanbul, Turkey.
- Francis Tyers, Felipe Sánchez-Martínez, Sergio Ortiz-Rojas, and Mikel Forcada. 2010. Free/open-source resources in the apertium platform for machine translation research and development. *The Prague Bulletin of Mathematical Linguistics* 93:67–76.
- Daniel Zeman, Filip Ginter, Jan Hajič, Joakim Nivre, Martin Popel, Milan Straka, and et al. 2017. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, pages 1–20.
- Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, Brussels, Belgium, pages 1–20.



**Jenna Kanerva & Filip Ginter & Sampo Pyysalo**  
**Turku Enhanced Parser Pipeline: From Raw Text to**  
**Enhanced Graphs in the IWPT 2020 Shared Task**

In Proceedings of the 16th International Conference on Parsing  
Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced  
Universal Dependencies. 2020.

# Turku Enhanced Parser Pipeline: From Raw Text to Enhanced Graphs in the IWPT 2020 Shared Task

Jenna Kanerva\* Filip Ginter Sampo Pyysalo  
TurkuNLP group, Department of Future Technologies  
University of Turku, Finland  
first.last@utu.fi

## Abstract

We present the approach of the TurkuNLP group to the IWPT 2020 shared task on Multilingual Parsing into Enhanced Universal Dependencies. The task involves 28 treebanks in 17 different languages and requires parsers to generate graph structures extending on the basic dependency trees. Our approach combines language-specific BERT models, the UDify parser, neural sequence-to-sequence lemmatization and a graph transformation approach encoding the enhanced structure into a dependency tree. Our submission averaged 84.5% ELAS, ranking first in the shared task. We make all methods and resources developed for this study freely available under open licenses from <https://turkunlp.org>.

## 1 Introduction

The Universal Dependencies<sup>1</sup> (UD) effort (Nivre et al., 2016, 2020) seeks to create cross-linguistically consistent dependency annotation and has to date produced more than 150 treebanks in 90 languages. UD is a broad and open community effort with more than 300 contributors (Zeman et al., 2019), and the resources they have created have been instrumental in driving progress in dependency parsing in recent years, also serving as the basis of widely attended CoNLL shared tasks on multilingual parsing in 2017 and 2018 (Zeman et al., 2017, 2018). While UD resources, the CoNLL shared tasks, and recent advances in deep learning-based parsing technology (Dozat et al., 2017; Kanerva et al., 2018; Kondratyuk and Straka, 2019) have contributed substantially to accurate dependency parsing using a consistent syntactic representation for a wide range of human languages, these efforts have focused almost exclusively on the *basic* UD dependency trees. UD defines also an

*enhanced* graph representation, which allows more detailed representation of the sentence. Common types of enhancements include null nodes for elided predicates, propagation of conjuncts for making connections between words more explicit, and augmentation of modifier labels with prepositional or case-marking information. The ability to produce enhanced UD graphs from raw text, previously explored by e.g. Schuster and Manning (2016), Nivre et al. (2018), and Schuster et al. (2018), would represent a further advance over existing tools.

The IWPT 2020 Shared Task on Multilingual Parsing into Enhanced Universal Dependencies<sup>2</sup> (Bouma et al., 2020) is the first shared task evaluation targeting the enhanced UD graph. The task was organized using data from 28 UD treebanks covering 17 languages, representing Baltic, Finnic, Germanic, Romance, Semitic, Slavic, and Southern Dravidian languages. We participated in the IWPT shared task with our parsing pipeline consisting of components for segmentation, part-of-speech and morphological tagging, lemmatization, dependency parsing, and enhanced dependency graph analysis. Our approach builds on custom pre-trained deep language models (Devlin et al., 2018), a deep neural network-based parser (Kondratyuk and Straka, 2019), a character-level sequence-to-sequence lemmatizer (Kanerva et al., 2020), and a custom graph transformation approach encoding an enhanced dependency graph in a labeled tree structure. The parsing pipeline is fully language agnostic, and therefore trainable with any UD treebank. Our submission to IWPT achieved an average enhanced labeled attachment score (ELAS) of 84.5%, the best performance among the 35 evaluated submissions from ten participating groups with an approximately 2% point margin to the second-best submission.

\*Equal contribution by all three authors

<sup>1</sup><https://universaldependencies.org/>

<sup>2</sup><https://universaldependencies.org/iwpt20/>

## 2 Shared Task Data

The shared task data involves 28 UD treebanks for 17 languages, representing the subset of treebanks for which enhanced dependencies are available. The enhanced dependencies fall into five types: gapping, propagation of conjuncts, controlled and raised subjects, relative clause antecedents, and case information. However, not all treebanks have all of these types. While the training data is divided according to individual treebanks, test data is divided on language level through pooling of the individual treebank test sets, without any direct possibility to identify which test set sentence originates from which source treebank. We note that this is a departure from previous UD parsing shared tasks, where the treebank distinction was preserved also in the test data. The training and development data range from less than 10,000 words for Tamil to over a million for Czech. Table 1 gathers statistics of the enhanced dependencies, compared to the base parse trees. We can see that the number of unique relation types increases by an order of magnitude, yet roughly 70-80% of the enhanced dependencies are copied unmodified from the base tree, and roughly 90-95% are a base dependency with its relation type modified.

## 3 System Overview

We next introduce our system and our approach to predicting enhanced dependencies.

### 3.1 Segmentation

For tokenization, multiword token expansion and sentence splitting we apply the Stanza toolkit by Qi et al. (2020) and its downloadable models trained on UD version 2.5 treebanks. Stanza implements a neural model that treats segmentation as a tagging problem over sequences of characters, where for a given character the model predicts whether it is the end of a token, the end of a sentence, or the end of a multiword token. Predicted multiword tokens are then expanded using a combination of a dictionary compiled from the training data and a sequence-to-sequence generation model.

### 3.2 Base Parser

We use the UDify dependency parser introduced by Kondratyuk and Straka (2019). UDify is a multi-task model for part-of-speech and morphological tagging, lemmatization and dependency parsing supporting fine-tuning of pre-trained BERT models

Treebank	Base	Enh	R%	UR%
Arabic-PADT	36	1074	66.1	92.9
Bulgarian-BTB	36	173	84.7	96.1
Czech-CAC	43	639	72.4	89.3
Czech-FicTree	42	295	78.7	90.5
Czech-PDT	43	759	75.6	91.8
Dutch-Alpino	35	416	83.3	95.7
Dutch-LassyS.	35	293	82.2	95.3
English-EWT	49	375	82.3	94.7
Estonian-EDT	38	560	76.1	98.3
Estonian-EWT	39	178	74.1	92.6
Finnish-TDT	45	418	74.1	91.1
French-Sequoia	46	71	93.9	95.3
Italian-ISDT	44	348	78.6	94.8
Latvian-LVTB	40	133	75.9	90.6
Lithuanian-A.	35	194	66.9	88.8
Polish-LFG	40	178	88.8	97.1
Polish-PDB	67	859	77.2	91.8
Russian-SynTag.	40	635	77.5	93.9
Slovak-SNK	41	268	81.0	94.3
Swedish-Talbank.	40	302	79.1	93.2
Tamil-TTB	28	116	69.3	97.3
Ukrainian-IU	57	351	77.5	91.6

Table 1: Statistics of base and enhanced relations from the training sections of the treebanks: *Base* is the number of unique relations in the base tree, *Enh* is the number of unique relations in the enhanced graph, *R%* is the proportion of enhanced dependencies also present in the base tree, and *UR%* is the proportion of unlabelled enhanced dependencies also present in the base tree. The letter R refers to recall.

on UD treebanks. UDify implements a multi-task network where a separate prediction layer for each task is added on top of the pre-trained BERT encoder. Additionally, instead of using only the top encoder layer representation in prediction, UDify adds attention vertically over the 12 layers of BERT, calculating a weighted sum of all intermediate representations of BERT layers for each token. All prediction layers as well as layer-wise attention are trained simultaneously, while also fine-tuning the pre-trained BERT weights.

In our shared task system we use UDify for part-of-speech tagging (UPOS), predicting morphological features (FEATS) as well as for dependency parsing. By contrast to the original UDify work, we train separate language-specific models rather than one model covering all languages.

### 3.3 Lemmatizer

For lemmatization we use the Universal Lemmatizer by Kanerva et al. (2020) trained on the shared task training data. The lemmatizer casts the task as a sequence-to-sequence rewrite problem where the input token is represented as a sequence of characters followed by a sequence of its part-of-speech

and morphological tags, and the desired lemma is then generated a character at time from the input. Following this approach, the contextual information needed for disambiguating between possible lemmas for ambiguous words is obtained directly from the predicted morphological tags, thus creating a compact context representation which generalizes well. In order to obtain predicted tags for lemmatization, we apply the lemmatizer as the final component in our pipeline.

### 3.4 Enhanced Representation

Since our base parser is only capable of reproducing trees, the enhanced representation needs to either be encoded into the base trees by enriching the set of dependency types, or alternatively introduced in a separate step after base parsing. In our system submission, we chose the former, but have also experimented with the latter approach. The overall approach of encoding the graph into a tree is well-known and has been applied previously, e.g. by a number of teams in the SemEval tasks on semantic dependency parsing (Open et al., 2014, 2015).

Our choices adhered to the following principles: (a) the LAS of the base parser must not be compromised, (b) the encoding must be language-independent and applicable to any treebank, and (c) the method must be sufficiently simple to be included in a production-grade parsing pipeline.

#### 3.4.1 Encoding into Base Tree

In order to encode enhanced dependencies into the base tree, we focused on a just four structures, which nevertheless cover the vast majority of the edges in the enhanced representation (see Table 2 below). The four structures and their encoding are shown in Figure 1. In the encoding, the base tree structure does not change; the enhanced relations are encoded into the base tree relations, also recording whether the enhanced dependency goes from or to the head in the base tree, or from or to the head of the head in the base tree. This encoding makes the decoding process straightforward and deterministic, because there can be at most one head and at most one head of head in the parse tree. The downside of this approach is that the number of unique relation types which the parser needs to predict increases substantially. Note that this encoding applies straightforwardly to cases where a token is the head or dependent in several enhanced relations; their encoding is simply concatenated.

The main reason for the increase in the num-

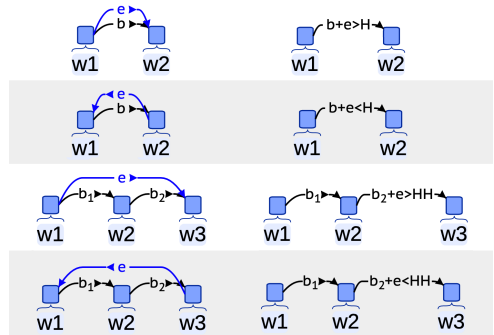


Figure 1: The four enhanced dependency structures currently captured in our encoding. The base (b) and enhanced (e) relations in the left column are encoded in a tree structure as in the right column. In the encoding, the symbol  $>$  stands for "relation from",  $<$  stands for "relation to",  $H$  is the head in the base tree, and  $HH$  is the head of the head in the base tree.

ber of unique relation types is the lexicalized relations which encode the lemma of a functional word (e.g. the *case* dependent) into the enhanced relation. To address this issue in a language-independent manner, we scan the enhanced relations for occurrences of a lemma of a dependent of the head or the dependent in the enhanced relation. If one is found, it is replaced with a placeholder encoding which position the lemma occurred at. For instance  $\{lemma-d-case\}$  indicates that this placeholder is to be replaced with the lemma of a case dependent of the dependent in this enhanced relation. Similarly,  $\{lemma-h-case\}$  indicates that this placeholder is to be replaced with the lemma of a case dependent of the head in this enhanced relation. Such delocalization is once again straightforward to reverse and in practice deterministic, although not so in theory, since a word can have several dependents of the same type.

The final feature of the enhanced representation that we address is the empty nodes occurring in elliptic constructions. Here, we once again rely on encoding of information into the base tree. The shared task evaluation procedure includes a step whereby empty nodes are removed and encoded in the form of enhanced relations that every two relations  $(h, e, r_1), (e, d, r_2)$  produce a new enhanced relation  $(h, d, r_1 > r_2)$  which encodes the presence of an empty node. Once all relations of the empty node are encoded in this manner, the empty node is removed. This representation is easy to reverse, and in practice allows one to reconstruct the empty



nodes in the enhanced representation except for their position in the sentence, which is not particularly relevant nor evaluated in the shared task. Only cases where a word has several empty node dependents with the same relation type cannot be reconstructed correctly.

The overall procedure for encoding the enhanced representation is:

1. Encode empty nodes as enhanced relations, remove from the graph
2. Replace all recognized function word lemmas with their corresponding placeholders
3. Encode all enhanced relations of the four types using the encoding in Figure 1, discard any other enhanced relations

This sequence of steps produces a tree representation that a standard dependency parser can be trained on. The output of the parser is decoded in the reverse order of the encoding steps, producing the enhanced representation. The decoding must take into account any errors the parser produced which might impair the decoding of the encoded representation, or produce an enhanced graph which does not validate as Universal Dependencies. In particular:

- Any relation headed by the root is given the type *root* regardless of the parser’s prediction.
- If a lemma placeholder cannot be reversed (e.g. when a parser predicts a placeholder  $\{lemma-d-case\}$  but there is no such dependent in the tree, the enhanced relation is discarded. Note that leads to unconnected words in the enhanced graph.
- Any word that remains unconnected in the enhanced graph is made the dependent of the same head, with the same relation, as in the base tree.
- For any (undirected) connected component that does not include the root node, we identify a word that all other words of the component can be reached from in the directed graph, and make this word a dependent of the root node. If no such word can be found, then the set of words with no incoming edge in the component are made dependents of the root node. This latter condition did not trigger in practice.

The encode–decode procedure can be evaluated by first encoding the enhanced training graphs into

Treebank	Rels	ELAS
Arabic-PADT	1,108	99.28
Bulgarian-BTB	152	99.22
Czech-CAC	939	98.13
Czech-FicTree	355	98.38
Czech-PDT	1,079	98.75
Dutch-Alpino	569	99.16
Dutch-LassySmall	420	99.23
English-EWT	611	98.89
Estonian-EDT	359	99.88
Estonian-EWT	202	99.74
Finnish-TDT	451	97.96
French-Sequoia	79	99.09
Italian-ISDT	561	99.53
Latvian-LVTB	405	97.94
Lithuanian-ALKSNIS	267	98.12
Polish-LFG	146	99.21
Polish-PDB	845	98.34
Russian-SynTagRus	1,119	99.57
Slovak-SNK	281	99.44
Swedish-Talbanken	494	99.16
Tamil-TTB	78	99.79
Ukrainian-IU	363	98.88

Table 2: Number of unique dependency relations after the encoding procedure, and the ELAS value after an encode–decode cycle. The latter number reflects to what extent the original enhanced graphs can be reconstructed after the encoding. The numbers are reported on the training portions of the treebanks.

trees, decoding back, and measuring the ELAS of the decoded data against the original. A lossless representation would result in ELAS of 100%. As shown in Table 2, this value is in the 97.9–99.9% range across all treebanks, meaning the encoding is not far from lossless, and only little gain can be expected from encoding more complex structures. Note, however, that this reflects the comparative structural simplicity of the enhanced relations present in the UD data, rather than the generality of our encoding. Table 2 also reports on the number of unique dependency relations in the training section of each treebank, showing an order of magnitude increase compared to the base tree.

### 3.4.2 Enhanced Relations as Tagging

The encoding of the enhanced relations into the base tree can also be seen as a tagging task, since every word has exactly one base relation, and therefore also exactly one relation in the encoded tree. It is therefore possible to first parse the sentence with a parser that predicts the base tree, and then subsequently tag the words with tags corresponding to the encoding of the enhanced relations, as introduced earlier, with the base parse tree serving as a source of features. The main advantage of such an approach would be guaranteeing that the

Model	Languages	References
Arabic-BERT	Arabic	<a href="https://github.com/alisafaya/Arabic-BERT">https://github.com/alisafaya/Arabic-BERT</a>
BERTje	Dutch	<a href="https://github.com/wietsedv/bertje">https://github.com/wietsedv/bertje</a> ; (de Vries et al., 2019)
BERT (original)	English	<a href="https://github.com/google-research/bert">https://github.com/google-research/bert</a> ; (Devlin et al., 2018)
FinBERT	Finnish	<a href="https://turkunlp.org/FinBERT/">https://turkunlp.org/FinBERT/</a> ; (Virtanen et al., 2019)
CamemBERT	French	<a href="https://camembert-model.fr/">https://camembert-model.fr/</a> ; (Martin et al., 2020)
Italian BERT	Italian	<a href="https://github.com/dbmdz/berts">https://github.com/dbmdz/berts</a>
RuBERT	Russian	<a href="https://github.com/deepmipt/deeppavlov/">https://github.com/deepmipt/deeppavlov/</a> ; (Kuratov and Arkhipov, 2019)
Slavic-BERT	Slavic <sup>1</sup>	<a href="https://github.com/deepmipt/Slavic-BERT-NER">https://github.com/deepmipt/Slavic-BERT-NER</a> ; (Arkhipov et al., 2019)
Swedish BERT	Swedish	<a href="https://github.com/Kungbib/swedish-bert-models">https://github.com/Kungbib/swedish-bert-models</a>
mBERT	104 lang.	<a href="https://github.com/google-research/bert">https://github.com/google-research/bert</a>

Table 3: Previously released BERT models for shared task languages. <sup>1</sup>Slavic-BERT is trained on Bulgarian, Czech, Polish, and Russian.

base LAS of the parser does not change, while the main disadvantage is the added complexity of an additional step and the possibility of error chaining.

We pursued this alternative approach in parallel to the main line of work. As the results presented in Section 5 show, however, the encoding of the enhanced dependencies does not negatively affect the base LAS, undermining the motivation for a separate tagging approach with its added software complexity. In our preliminary experiments on the development data, the tagging approach resulted in a minimally worse performance than the primary approach, and was therefore not pursued further.

## 4 Language Models

We apply transfer learning using pre-trained BERT models, using multilingual BERT<sup>3</sup> (mBERT) as a starting point. Based on recent studies introducing language-specific BERT models (Arkhipov et al., 2019; Virtanen et al., 2019; de Vries et al., 2019; Martin et al., 2020), we anticipated that parsing performance could be substantially improved by replacing the multilingual model with dedicated language-specific ones. To identify or create a model that would improve on performance with mBERT for every treebank in the shared task, we adopted a three-stage approach: 1) use previously released models, 2) pre-train a new model on Wikipedia data, and 3) continue pre-training on texts from a web crawl.

### 4.1 Previously Released Models

We considered the previously released models summarized in Table 3. Based on preliminary experiments, we focused on cased models in cases where both cased and uncased variants are available. We evaluated mBERT for all shared task treebanks,

Slavic-BERT for Bulgarian, Czech, Polish, and Russian, and the other models for treebanks for the individual languages that those models target.

### 4.2 Unannotated Texts

Our primary source of unannotated texts in various languages is Wikipedia. To extract plain text, we processed the full 2020/01/20 Wikipedia database backup dumps<sup>4</sup> for the various languages with WikiExtractor<sup>5</sup>. The basic statistics of extracted Wikipedia texts for the IWPT languages are summarized in Table 9 in the Appendix. We note that the sizes of these unannotated texts vary greatly between languages, ranging just over 20 million tokens for Latvian to nearly 3 billion for English. In many cases, languages with large Wikipedias also have large annotated treebanks, and vice versa; the language with the smallest amount of annotated training data in the shared task, Tamil, also ranks second from bottom in terms of the available unannotated Wikipedia data. We augmented the collection of unannotated texts for selected languages with texts drawn from OSCAR<sup>6</sup> (Ortiz Suárez et al., 2019), using unshuffled versions provided by the creators of the corpus (see Table 8 in the Appendix). The unshuffled version of the corpus is used since BERT training is carried out on text segments of up to 512 sub-words, far longer than most individual sentences. To reduce the level of noise in the web-crawled texts, we filtered the OSCAR source using 5-gram perplexity with a KenLM<sup>7</sup> language model estimated on Wikipedia data. In brief, we measured the average sentence-level perplexity  $t$  and filtered out any document where the average perplexity was greater than  $t$ . In terms of tokens, this procedure

<sup>4</sup><https://dumps.wikimedia.org/>

<sup>5</sup><https://github.com/attardi/wikiextractor>

<sup>6</sup><https://traces1.inria.fr/oscar/>

<sup>7</sup><https://github.com/kpu/kenlm>

<sup>3</sup><https://github.com/google-research/bert/blob/master/multilingual.md>

Treebank	Model	
	mBERT	Language-specific
Arabic PADT	<b>83.62</b>	82.76 (Arabic-BERT)
Bulgarian BTB	90.75	<b>91.83</b> (Slavic-BERT)
Czech CAC	91.80	<b>92.99</b> (Slavic-BERT)
Czech FicTree	92.31	<b>93.27</b> (Slavic-BERT)
Czech PDT	92.58	<b>93.44</b> (Slavic-BERT)
Dutch Alpino	92.58	<b>93.36</b> (BERTje)
Dutch LassySmall	<b>88.30</b>	87.69 (BERTje)
English EWT	90.08	<b>91.82</b> (BERT-large)
Estonian EWT	71.27	<b>73.08</b> (WikiBERT-et)
Finnish TDT	87.83	<b>92.89</b> (FinBERT)
French Sequoia	<b>93.12</b>	92.99 (CamemBERT)
Italian ISDT	92.75	<b>93.44</b> (Italian BERT)
Latvian LVTB	<b>86.71</b>	85.96 (WikiBERT-lv)
Lithuanian ALKSNIS	83.02	<b>85.26</b> (WikiBERT-lt)
Polish LFG	95.34	<b>96.22</b> (Slavic-BERT)
Polish PDB	91.90	<b>93.37</b> (Slavic-BERT)
Russian SynTagRus	92.06	<b>93.34</b> (RuBERT)
Slovak SNK	<b>92.52</b>	91.89 (WikiBERT-sk)
Swedish Talbanken	86.96	<b>90.56</b> (Swedish BERT)
Tamil TTB	<b>69.12</b>	67.38 (WikiBERT-ta)
Ukrainian IU	89.60	<b>91.25</b> (WikiBERT-uk)
Average	88.30	<b>89.28</b>

Table 4: UDify development set LAS performance with mBERT compared to language-specific BERTs

filtered out approx. 10% of the OSCAR data for Latvian and Slovak and 24% for Tamil.

### 4.3 Pre-training

For pre-training new BERT models, we largely follow the approach used to create the original BERT-base English model by Devlin et al. (2018). Specifically, we adapt the preprocessing pipeline and pre-training process introduced by Virtanen et al. (2019) for creating the Finnish BERT model. In brief, we train BERT-base models for 1M steps, the initial 900K with a maximum sequence length of 128 and the last 100K with 512, using the original BERT software<sup>8</sup> and the same optimizer parameters as Devlin et al. (2018) with the exception of batch size. Due to memory limitations, a batch size of 140 was used with 4 GPUs for the first 900K steps and a batch size of 20 with 8 GPUs for the last 100K steps. Nvidia V100 GPUs with 32 GB memory were used for pre-training. For comprehensive details of the preprocessing and pre-training process, we refer to the documentation of our pipeline.<sup>9</sup>

### 4.4 Language Model Evaluation

For evaluating pre-trained language models, we trained UDify with the shared task training data for

<sup>8</sup><https://github.com/google-research/bert>

<sup>9</sup><https://github.com/TurkuNLP/wikibert>

Treebank	Model	
	mBERT	Language-specific
Arabic PADT	83.62	<b>84.79</b> (WikiBERT-ar)
Dutch Alpino	92.58	<b>93.47</b> (WikiBERT-nl)
Dutch LassySmall	88.30	<b>89.23</b> (WikiBERT-nl)
French Sequoia	93.12	<b>93.21</b> (WikiBERT-fr)
Average	89.41	<b>90.18</b>

Table 5: UDify development set LAS performance with mBERT compared to additional WikiBERTs

Treebank	Model	
	mBERT	Language-specific
Latvian LVTB	86.71	<b>88.47</b> (Wiki+OSCAR-BERT-lv)
Slovak SNK	<b>92.52</b>	<b>92.52</b> (Wiki+OSCAR-BERT-sk)
Tamil TTB	69.12	<b>71.02</b> (Wiki+OSCAR-BERT-ta)
Average	82.78	<b>84.00</b>

Table 6: UDify development set LAS performance with mBERT compared to Wiki+OSCAR-BERTs

each language and evaluated on the corresponding development dataset using gold standard tokenization. The standard LAS metric was used to assess model performance.

Table 4 summarizes evaluation results comparing parsing performance with mBERT and language-specific models. As expected, we find that language-specific models outperform the multilingual model in most cases, averaging approximately 1% point higher LAS (~8% reduction in error). There are nevertheless a number of cases where UDify with mBERT outperforms the language-specific model. To address these cases, we introduced additional WikiBERT models for Arabic, Dutch, and French. Results comparing the performance of these models with mBERT are summarized in Table 5. We find that in each case using the WikiBERT model improves on results with mBERT, with absolute differences around 1% point for the Arabic and Dutch treebanks but very limited (~0.1% point) difference for French, averaging 0.8% point higher LAS than mBERT (~7% reduction in error).

Finally, there are three languages for which no previously released language-specific model was available and the WikiBERT failed to improve on performance with mBERT: Latvian, Slovak, and Tamil. For these languages, we continued pre-training with texts from OSCAR for an additional 300,000 steps. Table 6 summarizes performance with these models. For Slovak, the new model improves over the WikiBERT model performance but merely matches the performance with mBERT, while the Latvian and Tamil models outperform

Language	Team									
	adapt	clasp	emory	fastparse	koeksala	orange	robert	shanghai	turku	unipi
Arabic	57.19	51.26	67.26	66.92	60.84	70.96	0.0	63.41	<b>77.82</b>	57.79
Bulgarian	77.29	84.90	88.19	84.86	68.88	89.42	0.0	78.67	<b>90.73</b>	84.93
Czech	66.41	67.13	85.51	77.21	61.11	86.95	0.0	75.43	<b>87.51</b>	75.99
Dutch	67.67	78.93	80.72	77.37	62.93	<b>85.14</b>	0.0	70.94	84.73	77.62
English	70.44	82.87	85.30	78.45	65.37	85.21	<b>88.94</b>	72.34	87.15	83.95
Estonian	61.12	60.44	81.36	74.09	59.07	81.03	0.0	74.91	<b>84.54</b>	57.24
Finnish	72.37	65.96	82.96	75.73	67.54	86.24	0.0	75.99	<b>89.49</b>	72.13
French	74.74	72.76	<b>86.23</b>	77.77	67.93	83.63	0.0	76.99	85.90	78.85
Italian	71.98	87.14	88.52	84.77	69.08	90.83	0.0	73.08	<b>91.54</b>	89.14
Latvian	72.41	66.01	79.19	75.57	64.75	82.11	0.0	77.77	<b>84.94</b>	68.23
Lithuanian	58.36	52.56	66.12	61.41	56.28	75.89	0.0	66.85	<b>77.64</b>	61.06
Polish	65.86	71.22	82.39	74.54	61.34	80.39	0.0	71.01	<b>84.64</b>	70.61
Russian	75.27	70.37	88.60	80.35	64.23	89.84	0.0	78.26	<b>90.69</b>	76.90
Slovak	68.43	65.16	82.72	73.46	64.08	84.36	0.0	73.14	<b>88.56</b>	81.40
Swedish	68.39	71.35	78.19	75.24	64.50	83.27	0.0	69.60	<b>85.64</b>	78.73
Tamil	48.47	42.15	54.26	46.99	47.44	<b>64.23</b>	0.0	48.20	57.83	48.50
Ukrainian	66.43	63.24	79.69	74.02	64.17	84.64	0.0	72.98	<b>87.22</b>	73.90
Average	67.23	67.85	79.84	74.04	62.91	82.60	5.23	71.74	<b>84.50</b>	72.76

Table 7: ELAS results for submissions to IWPT 2020 shared task. Team names abbreviated for space: emory = emorynlp, orange = orange\_deskin, robert = robertnlp, shanghai = shanghaitech\_alibaba, turku = turkunlp.

mBERT with a nearly 2% point absolute difference in LAS. On average, the new models improve on mBERT by 1.2% points, again an approx. 7% reduction in error.

## 5 Results

For our final submission, we trained a model for each language using the largest treebank (in terms of token count) for the language in the shared task data release. All segmentation, tagging, parsing, and lemmatization models are thus monolingual and trained using only a single treebank. Each UDify model is fine-tuned for 160 epochs using a number of warm-up steps<sup>10</sup> roughly equal to a single pass over the training dataset. For each language the fine-tuning is based on a custom pre-trained BERT model selected as detailed in Section 4.4. Lemmatization models do not require any external resources, and all hyperparameters follow the values used in Kanerva et al. (2020).

The primary evaluation metric in the shared task is ELAS (Labeled Attachment Score on Enhanced dependencies), which calculates F-score over the set of enhanced dependencies in the system output and gold standard.<sup>11</sup> Table 7 summarizes the ELAS results for all ten teams participating the shared task. We note that in addition to achieving

the best average ELAS performance, our system also outperforms all other submissions for 13 out of the 17 individual languages included in the task. For these 13 languages, the largest absolute differences for the second-best result are for Arabic (~6.9% points), Slovak (~4.2% points), Estonian, and Finnish (both slightly above 3% points).

For the four languages where our system did not achieve the highest ELAS results, the differences to the highest-performing submission are small (0.3-0.4% points) for Dutch and French, and 1.8% points for English. However, there is a more than 6% point difference to the top result for Tamil, the language with the smallest treebank in the shared task. This difference indicates a tradeoff of our approach in training monolingual models: languages with particularly limited resources do not gain support from annotations in other languages as they would in multilingual training.

Table 10 in the Appendix shows average results for all metrics excepting for XPOS, which due time limitations we decided not to predict, and AllTags, which is not meaningfully defined when not predicting XPOS. We note that our system achieves the best performance for all but two metrics, outperforming other systems in segmentation (Tokens, Words, Sentences), part-of-speech tagging (UPOS), lemmatization (Lemmas) as well as for all but one of the seven dependency attachment score (\*AS) metrics. Our system falls behind the best-performing submission (orange\_deskin) for the UFeats and MLAS metrics. As MLAS (Morphology-Aware Labeled Attachment Score)

<sup>10</sup>During warm-up, the learning rate is gradually increased from zero to its initial value, so as to avoid large changes at the very beginning of the training.

<sup>11</sup>Note that in UD many of the base layer relations are repeated in the enhanced graph, and therefore the ELAS metric evaluates a combination of basic dependencies and enhancements as seen in statistics presented in Table 1.

requires selected features to match, the results for these two metrics likely both reflect performance for morphological features. The absolute difference of our system to the top result for UFeats is 1.2% points, reflecting a 20% relative increase in error and indicating a clear remaining point for improvement in our system.

## 6 Discussion

Cross-lingual compatibility is a major goal of the UD effort and the ability to train multilingual models where lower-resourced languages can benefit from data in higher-resourced languages a clearly desirable aim in language modeling. While our approach – which trains monolingual models and uses language-specific pre-trained models – can be seen as running counter to these goals, we do nevertheless share them. Our choice to train separate models for each language for the shared task is based in part in awareness of remaining compatibility issues in UD treebanks, even within languages. We hope contrasting results for joint and language-specific models for this shared task will help identify and resolve some of these challenges. Regarding multilingual language models, we note that in aiming to cover more than 100 languages without a corresponding increase in model and vocabulary size, mBERT faces multiple challenges in its capacity, and the model training does not fully balance lower- and higher-resourced languages. While we here found language-specific models to outperform a specific mBERT model, highly multilingual models addressing these challenges might well be competitive with language-specific ones, and the creation of such models would greatly benefit practical parsing efforts targeting a large number of languages.

To study the impact of the language-specific language models in our shared task results, we reproduce our pipeline using exactly same configurations except for replacing all language-specific BERT models with the multilingual mBERT. In this experiment, all languages are using the same multilingual language model as a starting point, later individually fine-tuned for each language while training the language-specific parsing models. When comparing these models to the official submissions of all 10 teams, the average ELAS is approximately 1.7% points below our own primary submission (~11% increase in error), but still slightly above the second best submission by approximately 0.2%

points. This means that, our pipeline would have reached the highest average ELAS score among the official submissions also without the language-specific BERT models, but only with a very thin margin to the next best team.

## 7 Conclusions

We have presented the approach of the TurkuNLP group to the IWPT 2020 shared task on Multilingual Parsing into Enhanced Universal Dependencies. Our approach is based on deep transfer learning with language-specific models, the state-of-the-art UDify neural parsing pipeline, sequence-to-sequence lemmatization, and a graph transformation approach to predicting enhanced dependency graphs. Our submission to the shared task achieved the highest performance for the primary evaluation metric (ELAS) both on average as well as for 13 out of the 17 languages involved in the task, also achieving the highest average performance for most other evaluation metrics.

All of the methods and resources developed for this study are made freely available under open licenses from <https://turkunlp.org>.

## Acknowledgments

We gratefully acknowledge the support of the Academy of Finland, and CSC — the Finnish IT Center for Science for providing computational resources. We also thank the creators of the OSCAR corpus for making unshuffled versions of their corpus available for this work.

## References

- Mikhail Arkhipov, Maria Trofimova, Yurii Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93.
- Gosse Bouma, Djamé Seddah, and Daniel Zeman. 2020. Overview of the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, Seattle, US. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

- Timothy Dozat, Peng Qi, and Christopher D Manning. 2017. Stanford’s graph-based neural dependency parser at the CoNLL 2017 Shared Task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30.
- Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku neural parser pipeline: An end-to-end system for the CoNLL 2018 Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 133–142.
- Jenna Kanerva, Filip Ginter, and Tapio Salakoski. 2020. [Universal Lemmatizer: A sequence to sequence model for lemmatizing Universal Dependencies treebanks](#). *Natural Language Engineering*. To appear.
- Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.
- Yuri Kuratov and Mikhail Arhipov. 2019. Adaptation of deep bidirectional multilingual transformers for Russian language. *arXiv preprint arXiv:1905.07213*.
- Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villamonte de la Clergerie, Djamel Seddah, and Benoît Sagot. 2020. CamemBERT: a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 1659–1666.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4027–4036. European Language Resources Association.
- Joakim Nivre, Paola Marongiu, Filip Ginter, Jenna Kanerva, Simonetta Montemagni, Sebastian Schuster, and Maria Simi. 2018. Enhancing Universal Dependency Treebanks: A Case Study. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 102–107. Association for Computational Linguistics.
- Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Uřešová. 2015. SemEval 2015 Task 18: Broad-Coverage Semantic Dependency Parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926. Association for Computational Linguistics.
- Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 Task 8: Broad-Coverage Semantic Dependency Parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72. Association for Computational Linguistics.
- Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), System Demonstrations*.
- Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 2371–2378. European Language Resources Association (ELRA).
- Sebastian Schuster, Joakim Nivre, and Christopher D. Manning. 2018. Sentences with Gapping: Parsing and Reconstructing Elided Predicates. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1156–1168, New Orleans, Louisiana. Association for Computational Linguistics.
- Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for Finnish. *arXiv preprint arXiv:1912.07076*.
- Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. *arXiv preprint arXiv:1912.09582*.
- Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and

Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 1–21.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Noëmi Aeppli, Željko Agić, Lars Ahrenberg, Gabrielè Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnè Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çoltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomáš Erjavec, Aline Etienne, Wograine Evelyn, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökürmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Gričiūtė, Matias Groni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỳ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Oľáždé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettererová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Kamil Kopacewicz, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê H òng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Maria Livovina, Yuan Li, Nikola Ljubešić, Olga Logi-

nova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Măranduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskiy, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horňáček, Anna Nedoluzhko, Gunta Nešpore-Bėrkalne, Luong Nguyễn Thĩ, Huyễn Nguyễn Thĩ Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Mai Olúòkun, Adédayoand Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Lapińska, Siyao Peng, Cemel-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Riebler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roşca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamel Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibus-sirri, Dmitry Sichinava, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Takaaki Tanaka, Isabelle Tellier, Guillaume Thomas, Lisi Torga, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uri, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilian Wendt, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir

Zeldes, Manying Zhang, and Hanzhi Zhu. 2019. [Universal Dependencies 2.5](#). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajič jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Mäsilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Drogonova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. [CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies](#). In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19. Association for Computational Linguistics.



## A Appendix

Table 8 shows the same statistics for the OSCAR corpora of selected languages, and Table 9 summarizes the basic statistics of extracted Wikipedia texts for the IWPT languages. Table 10 shows average results for various metrics for all submissions to IWPT 2020 shared task.

Language	Docs	Sents	Tokens	Chars
Latvian	1.6M	34M	628M	4.0B
Slovak	5.5M	99M	1.5B	9.1B
Tamil	1.3M	39M	528M	3.8B

Table 8: OSCAR source statistics for selected IWPT 2020 shared task languages

Language	Docs	Sents	Tokens	Chars
Arabic	1.0M	8.0M	184M	889M
Bulgarian	259K	4.1M	71M	397M
Czech	444K	7.9M	143M	804M
Dutch	2.0M	19M	300M	1.7B
English	5.9M	124M	2.7B	14B
Estonian	205K	2.7M	38M	252M
Finnish	477K	7.4M	97M	731M
French	2.2M	34M	858M	4.5B
Italian	1.6M	22M	579M	3.0B
Latvian	99K	1.3M	21M	126M
Lithuanian	196K	2.3M	34M	207M
Polish	1.4M	16M	282M	1.7B
Russian	1.6M	31M	565M	3.5B
Slovak	232K	2.8M	39M	229M
Swedish	3.7M	30M	364M	2.1B
Tamil	132K	1.9M	26M	195M
Ukrainian	979K	15M	260M	1.5B

Table 9: Wikipedia source statistics for IWPT 2020 shared task languages

Metric	Team									
	adapt	clasp	emory	fastparse	koepsala	orange	robert	shanghai	turku	unipi
Tokens	99.54	99.72	99.66	99.66	99.66	99.68	5.85	99.67	<b>99.74</b>	99.63
Words	98.96	99.12	99.06	99.06	99.06	99.09	5.85	99.08	<b>99.13</b>	99.03
Sentences	89.22	92.34	91.25	91.18	91.25	90.24	5.07	91.97	<b>92.41</b>	90.56
UPOS	95.88	95.48	93.63	93.60	93.63	96.69	5.63	0.63	<b>96.75</b>	92.78
UFeats	91.36	90.66	87.35	88.11	87.35	<b>93.98</b>	5.57	32.84	92.77	86.02
Lemmas	95.40	95.15	92.30	92.23	92.30	95.80	5.62	0.02	<b>95.96</b>	91.35
UAS	87.18	86.41	88.95	82.55	79.97	89.45	5.26	13.01	<b>89.92</b>	84.90
LAS	84.09	82.66	86.14	77.57	75.41	86.79	5.11	0.99	<b>87.31</b>	80.74
CLAS	81.56	79.66	83.81	72.97	71.18	84.42	5.00	1.22	<b>85.23</b>	77.42
MLAS	72.57	69.55	67.84	60.82	60.54	<b>77.75</b>	4.51	0.01	76.63	62.73
BLEX	78.11	76.00	76.11	66.70	65.38	80.86	4.73	0.00	<b>81.93</b>	70.03
EULAS	69.42	80.18	81.26	75.96	64.93	84.62	5.26	73.01	<b>85.83</b>	78.82
ELAS	67.23	67.85	79.84	74.04	62.91	82.60	5.23	71.74	<b>84.50</b>	72.76

Table 10: Average results for different metrics for submissions to IWPT 2020 shared task. Team names abbreviated for space: emory = emorynlp, orange = orange\_deskin, robert = robertnlp, shanghai = shanghaitech\_alibaba, turku = turkunlp.




**Jenna Kanerva & Filip Ginter & Li-Hsin Chang & Iiro Rastas  
& Valtteri Skantsi & Jemina Kilpeläinen & Hanna-Mari Kupari  
& Aurora Piirto & Jenna Saarni & Maija Sevón & Otto Tarkka  
Towards Diverse and Contextually Anchored Paraphrase  
Modeling: A Dataset and Baselines for Finnish**

Natural Language Engineering. Published online 2023; 1-35.



ARTICLE

# Towards diverse and contextually anchored paraphrase modeling: A dataset and baselines for Finnish

Jenna Kanerva\* , Filip Ginter, Li-Hsin Chang, Iiro Rastas, Valtteri Skantsi, Jemina Kilpeläinen, Hanna-Mari Kupari, Aurora Piirto, Jenna Saarni, Maija Sevón and Otto Tarkka

TurkuNLP, Department of Computing, University of Turku, Turku, Finland

\*Corresponding author. Email: [jmnybl@utu.fi](mailto:jmnybl@utu.fi)

(Received 1 July 2022; revised 21 December 2022; accepted 17 January 2023)

## Abstract

In this paper, we study natural language paraphrasing from both corpus creation and modeling points of view. We focus in particular on the methodology that allows the extraction of challenging examples of paraphrase pairs in their natural textual context, leading to a dataset potentially more suitable for evaluating the models' ability to represent meaning, especially in document context, when compared with those gathered using various sentence-level heuristics. To this end, we introduce the Turku Paraphrase Corpus, the first large-scale, fully manually annotated corpus of paraphrases in Finnish. The corpus contains 104,645 manually labeled paraphrase pairs, of which 98% are verified to be true paraphrases, either universally or within their present context. In order to control the diversity of the paraphrase pairs and avoid certain biases easily introduced in automatic candidate extraction, the paraphrases are manually collected from different paraphrase-rich text sources. This allows us to create a challenging dataset including longer and more lexically diverse paraphrases than can be expected from those collected through heuristics. In addition to quality, this also allows us to keep the original document context for each pair, making it possible to study paraphrasing in context. To our knowledge, this is the first paraphrase corpus which provides the original document context for the annotated pairs.

We also study several paraphrase models trained and evaluated on the new data. Our initial paraphrase classification experiments indicate a challenging nature of the dataset when classifying using the detailed labeling scheme used in the corpus annotation, the accuracy substantially lacking behind human performance. However, when evaluating the models on a large scale paraphrase retrieval task on almost 400M candidate sentences, the results are highly encouraging, 29–53% of the pairs being ranked in the top 10 depending on the paraphrase type. The Turku Paraphrase Corpus is available at [github.com/TurkuNLP/Turku-paraphrase-corpus](https://github.com/TurkuNLP/Turku-paraphrase-corpus) as well as through the popular HuggingFace datasets under the CC-BY-SA license.

**Keywords:** Paraphrasing; Corpus annotation; Finnish; Paraphrase modeling

## 1. Introduction

Restating the same meaning in different wording, that is paraphrasing, occurs naturally in human communication, either by the same speaker repeating the message multiple times with different words, or by multiple speakers conveying the same message in different places. While a strict definition of a paraphrase requires the two statements to convey exactly the same meaning, often in natural language processing (NLP) and computational linguistics studies some form of a practical definition is adopted, requiring only having approximately the same meaning. The degree to

which the strict definition is relaxed differs across the various works that address paraphrasing (Bhagat and Hovy 2013).

In NLP, paraphrasing poses interesting challenges in the context of different natural language understanding and generation tasks such as machine translation, machine reading, plagiarism detection, question answering, and textual entailment (Mehdizadeh Seraj, Siahbani, and Sarkar 2015; Altheneyan and Menai 2019; Soni and Roberts 2019). The large, pre-trained language models that have recently become the methodological backbone of NLP have brought about a distinct shift towards more meaning-oriented tasks for model fine-tuning and evaluation. A typical example of such language understanding tasks is entailment detection, with the paraphrase task raising in interest recently, naturally depending on the availability of datasets for the task. Existing paraphrase corpora are typically either large and automatically constructed, or relatively small and manually annotated. Whereas manually annotated corpora are often too small for language model fine-tuning, automatically gathered larger datasets may introduce unwanted bias towards shorter paraphrases with higher lexical similarity due to the corpus-creation heuristics. Moreover, the manually annotated examples are often, although not always, sampled from a larger set of automatically gathered set of examples, carrying over any biases present in the automatic selection heuristics. In view of this situation, there is a need for paraphrase corpora of suitable size for language model fine-tuning, with high quality paraphrases that facilitate language understanding without reliance on surface lexical cues.

In this work, we set out to create a paraphrase corpus for Finnish, specifically aiming at producing a dataset not biased towards simple pairs that can be identified through a simple heuristic. Further, we aim to create a dataset sufficient in size for model training. Our primary motivation is to equip Finnish NLP for research and applications in natural language understanding.

To this end, we develop and apply an extraction protocol for manually collecting text segments that constitute true paraphrases from different paraphrase-rich text sources. Seeing that manual effort is best focused on searching for positive examples of paraphrases, we use automatic extraction of negative paraphrase candidates so as to obtain a dataset suitable for paraphrase classification model training. The concentration of effort on collecting true paraphrases strives for effective usage of the annotation person-months, as nonparaphrases can be more easily collected automatically. In addition, it is a more clearly defined task for the annotators to extract “paraphrases” than to extract “related segments that are not paraphrases”.

Importantly, during the manual paraphrase extraction, the position of the statement in the original source document is stored together with the extracted paraphrase pairs, allowing us to evaluate paraphrases in their natural document context, distinguishing between paraphrases in the given context compared with all possible contexts. To our knowledge, this property sets our work apart from other paraphrase corpora, as it is the first large-scale corpus of sentential paraphrases including manual paraphrase candidate extraction or document context information for the paraphrase pairs.

Together with the dataset, we also examine several paraphrase models trained on the data, as well as include a large-scale paraphrase mining evaluation, where we test how accurately the paraphrase models are able to identify the correct paraphrase pairs when hidden among almost 400M candidate sentences.

The paper is structured as follows. First, we describe the related work in paraphrasing in Section 2. In Sections 3, 4 and 5, we present the overall annotation workflow separated into three phases: heuristic retrieval of related document pairs from different text sources, manual paraphrase candidate extraction from these document pairs, and manual annotation of the extracted candidates. In Section 6, we present the corpus statistics and evaluation, and in Sections 7 and 8, we describe the semi-automatic methods for extracting closely related but negative paraphrase candidates and provide experimental results on both paraphrase classification as well as on paraphrase mining.

## 2. Related work

Several paraphrase corpora exist, greatly varying in terms of size, extraction methods used, and whether and to what degree the paraphrase pairs undergo manual verification. While most of the paraphrasing studies are carried out on English, paraphrase corpora exist for other languages as well. In addition, a few multilingual paraphrase resources exist. Next, we will review the most relevant work on building paraphrase resources.

### 2.1. Paraphrase datasets for English

There are numerous English paraphrase datasets in existence. Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett 2005) contains 5.8K paraphrase pairs automatically extracted from an online news collection. Heuristics to identify candidate document pairs and candidate sentences from the documents are used for the extraction, followed by filtering by classifier and finally manual binary annotation using labels (paraphrase or not). Twitter URL Corpus (TUC) (Lan *et al.* 2017) is a collection of 52K paraphrase pairs extracted based on shared URLs in news-related tweets. All pairs are manually labeled to be either paraphrases or nonparaphrases. ParaSCI (Dong, Wan, and Cao 2021) contains 350K automatically extracted paraphrase candidates from ACL and arXiv papers. The extraction heuristics consider term definitions, citation information, and sentence embedding similarity. The paraphrase candidates are automatically filtered without manual labels. ParaNMT-50M (Wieting and Gimpel 2018) contains over 50M sentential paraphrase candidates automatically generated by machine translating the Czech sentences from Czech-English parallel corpora to English. PARADE (He *et al.* 2020) is a collection of 10K paraphrase pairs collected from online user-generated flashcards for computer science related concepts. Definitions for a given term are clustered before in-cluster candidate extraction to reduce candidate selection noise. The candidate examples are subsequently manually assigned labels based on a four-label scheme. Quora Question Pairs (QQP)<sup>a</sup> is a collection of question headings from the Quora forum marked with either duplicate or not. Though the QQP dataset is comparatively large (404K pairs) and includes manual labels, the labeling is not originally intended for paraphrasing nor guaranteed to be perfect by the dataset providers. Additionally, Federmann, Elachqar, and Quirk (2019) evaluated different methods for paraphrase dataset generation on 500 English source sentences. These methods include monolingual human paraphrasing as well as translation roundtrip using both human and machine translation on different intermediate languages, but unfortunately the resulting dataset does not seem to be publicly available.

### 2.2. Other monolingual datasets

Monolingual paraphrase datasets have been constructed for many languages other than English, for instance Chinese, Japanese, Punjabi, Russian, and Turkish. The Phoenix Paraphrasing Dataset,<sup>b</sup> released by Baidu, consists of 500K Chinese paraphrase candidates that are short segments of queries. The dataset is created by first collecting seed paraphrase candidates to train a model, which is then used to generate more candidates. The generated pairs are subsequently filtered by a paraphrase recognition model. Shimohata *et al.* (2004) build a Japanese paraphrase corpus containing 683 paraphrase pairs to simplify long spoken-language sentences into machine translation-suitable forms. The paraphrases are travel conversations and their human-paraphrased versions. The paraphrasing strategies are removal of unnecessary redundancy, segmentation of long sentences, and summarization. Arwinder Singh (2020) automatically create a paraphrase dataset for Punjabi with phrasal and sentential paraphrase candidates. They

<sup>a</sup><https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs>.

<sup>b</sup><https://ai.baidu.com/broad/introduction?dataset=paraphrasing>.

cluster news headings and articles on the same event from the same day and extract paraphrase candidates with high vector similarity. Nearly 115K phrasal and 75K sentential paraphrase candidates are automatically collected. Manual binary categorization of 1000 pairs from each type shows 88% accuracy for phrasal and 70% for sentential paraphrase candidates. ParaPhraser (Pivovarov *et al.* 2018) is a Russian corpus created through automatic candidate extraction of news headlines from Russian news agencies followed by crowd-sourced manual annotation. It includes over 7K paraphrase pairs classified into nonparaphrases, near-paraphrases, and precise-paraphrases. Due to it not being of sufficient size for text generation, the ParaPhraser Plus dataset (Gudkov, Mitrofanova, and Filippikh 2020) has been gathered to enable text generation, with over 56M sentential paraphrase candidates. ParaPhraser Plus is created by automatically clustering news headlines by events over a 10-year period and enumerating all pairs of sentences in a cluster. The Turkish Paraphrase Corpus (TuPC) (Eyecioglu and Keller 2018) contains 1002 paraphrase pairs hand-picked from a pool of automatically paired sentences. The automatic pairing involves all-by-all sentence comparison and heuristic filtering based on length and word overlap of sentences from crawled news articles. All selected sentences are manually assigned a numeric label between 0 and 5 quantifying their degree of paraphrase.

### 2.3. Multilingual paraphrase datasets

Automatic paraphrase recognition oftentimes relies on language pivoting of multilingual parallel datasets. Pivoting is based on the assumption that identical translation possibly entails a paraphrase, and thus use sentence alignments to recognize potential different surface realizations of an identical or near-identical translation. Multilingual paraphrase datasets automatically extracted by language pivoting include Opusparcus (Creutz 2018) and TaPaCo (Scherrer 2020). Opusparcus (Creutz 2018) contains paraphrases for 6 languages and TaPaCo (Scherrer 2020) 73 languages, both including also a Finnish subsection. Opusparcus contains automatically extracted candidate paraphrases from alternative translations of movie and TV show subtitles. While all of the paraphrase candidates are automatically extracted, each language has a manually annotated subset of a few thousand paraphrase pairs. TaPaCo consists of paraphrase candidate pairs automatically extracted from the Tatoeba dataset,<sup>c</sup> a multilingual crowd-sourced database of sentences and translations thereof. The paraphrase candidates are assigned into “sets” rather than pairs, and sentences in a set are considered paraphrases of one another. The dataset does not have any manual annotation. Another multilingual paraphrase collection also extracted through language pivoting is Paraphrase Database (PPDB) (Ganitkevitch, Van Durme, and Callison-Burch 2013). Unlike the previously mentioned corpora containing sentential paraphrase candidates, PPDB include only lexical, phrasal, and syntactic paraphrase candidates collected automatically. PPDB has an English collection and a multilingual expansion that includes Finnish (Ganitkevitch and Callison-Burch 2014); however, most of the Finnish candidates in PPDB are just different inflectional variants of the same lexical items.

### 2.4. Resources for Finnish

The Turku Paraphrase Corpus introduced in this paper, the first large-scale, manually annotated paraphrase corpus for Finnish, includes 91,604 manually extracted and labeled paraphrases with an additional 13,041 human-made rephrasing of statements. While the first incomplete version of the corpus was released in Kanerva *et al.* (2021b), the current work extends the contributions into multiple directions: (1) the corpus size is doubled from the first release, (2) the text sources used to gather the paraphrases are extended from alternative subtitles and news headings to include also news articles, university student essays, translation exercises made by university students, as

<sup>c</sup><https://tatoeba.org/eng/>.

well as messages from an online discussion forum, (3) each manually extracted paraphrase is distributed together with the original document context to allow studies on paraphrasing in context, (4) in addition to manually extracted and labeled paraphrases, an automatically extracted subset of the corpus that contains related nonparaphrase segments is provided to support paraphrase classification.

Apart from the early release of the Turku Paraphrase Corpus, prior to this work only two resources of sentential paraphrases were available for Finnish, the two multilingual datasets Opusparcus and TaPaCo as mentioned above. Opusparcus dataset provides 3700 manually annotated paraphrase pairs for Finnish with an additional release of automatically scored and filtered candidates with different quality threshold ranging from 480K to few million candidates. TaPaCo dataset includes 12K paraphrase candidates for Finnish without any manual verification. A more detailed comparison of these two datasets and our corpus is given in Section 6.2.

### 3. Text sources for paraphrase extraction

One of the core questions we set out to address in this work is that of bias in paraphrase candidate selection. Here, we specifically want to avoid using heuristics as an initial candidate selection step so as to ensure that the resulting dataset also contains “difficult” examples that would be missed by heuristic selection. To this end, we rely on manual paraphrase extraction, where an annotator receives two related text documents presented alongside each other, and extracts all segments which can be considered as nontrivial paraphrases from the document pair (more details of the actual extraction work is given in Section 4.1). Therefore, in order to obtain sufficiently many paraphrases for the person-months we are able to spend, the text sources used in manual extraction need to be paraphrase-rich, that is have a high probability for naturally occurring paraphrases. Such text sources include for example independently written news articles reporting on the same event, alternative translations of the same source material, different student essays and exam answers to the same assignment, related questions with their replies in discussion fora, and other sources where one can assume different writers using distinct wording to state similar meanings.

We aim to strike a balance between sampling as many text sources as possible, optimizing the usage of person-months available for annotation, and the practical need to reach the goal of 100,000 paraphrase pairs set in the project plan based on which this work was funded. We utilize five different text sources: (1) alternative Finnish subtitles for the same movies or TV episodes, (2) news headings and articles discussing the same event in two different Finnish news sites, (3) different messages with identical title and sub-forum information from a popular Finnish discussion forum, (4) alternative student translations from university translation courses, and (5) student essays answering the same question in university course exams. Next, each text source is described separately introducing the specific methods used to select related document pairs for manual paraphrase candidate extraction.

#### 3.1. Alternative subtitles

OpenSubtitles<sup>d</sup> provides a large, vastly multilingual collection of user-contributed subtitles for various movies and TV episodes. The subtitles are available in a large number of languages, and oftentimes there are same-language alternative subtitles for a single movie/episode created independently. These can be viewed as independent translations of the same underlying content and offer an opportunity to make use of the natural variation therein. Through comparing, side-by-side, two alternative subtitle versions of a single movie or TV episode, many naturally occurring paraphrases are likely to be found.

<sup>d</sup><http://www.opensubtitles.org>.



We selected all movies and TV episodes with at least two alternative subtitle versions in Finnish from the database dump of OpenSubtitles2018 obtained through the OPUS corpus (Tiedemann 2012). We measure lexical similarity of alternative subtitle versions by TF-IDF weighted document vectors based on character *n*-grams extracted from within word boundaries. We exclude document pairs with too low or too high document vector cosine similarity values, so as to filter out document pairs with low interesting paraphrase candidate density. This is because a very high similarity often reflects identical subtitles with formatting differences, whereas a very low similarity tends to stem from misalignments caused by incorrect identifiers in the source data and other problems in the data. After this exclusion, the most lexically distant pair is used for paraphrase extraction if there are more than two versions available. For each movie/episode, the two selected subtitle versions are approximately aligned line-by-line using the subtitle timestamps. As we strive to collect paraphrase candidates from as diverse sources as possible, we divide each movie or episode into 15-minute-long segments. For each movie or TV episode, only one or two random segments are used to extract paraphrase candidates. The random selection is intended to prevent accidentally biasing the selection towards typical language used in the beginning of a story.

Altogether, we obtained aligned alternative subtitles for 1700 unique movies and TV series, demonstrating that alternative subtitle versions are surprisingly prolific in OpenSubtitles. We consider movies to be unique items, while episodes from TV series are considered mutually related due to their overlap in plot and characters, resulting in an overlapping in topic and language. After a period of initial annotation, we noticed a topic bias towards certain TV series with large numbers of episodes. We therefore adjusted the number of annotated episodes to be 10 at the highest from each TV series in all subsequent annotation. In total, over 2700 individual movies and TV episodes were used in the corpus construction. Ideally, only one 15-minute segment from each movie or TV episode would be used for candidate extraction, but due to not having enough other paraphrase-rich sources, we conducted a second round of candidate extraction where a second random segment is used after all available movie and TV episodes had been gone through once. The 1300 movies and TV episodes used in the second round were selected based on the number of paraphrase candidate pairs extracted in the first round, the higher the number, the higher the precedence a movie is assigned. In the end, approximately 4100 15-minute-long subtitle segment pairs were used in the corpus construction.

### **3.2. News articles and headings**

We have downloaded news articles through open RSS feeds of different Finnish news sites during 2017–2020, resulting in a substantial collection of news from numerous complementary sources. For the corpus creation, we narrow the data down to two sources: the Finnish Broadcasting Company (YLE) and Helsingin Sanomat (HS, English translation: Helsinki News). The news are aligned using a 7-day sliding window on time of publication, combined with cosine similarity of TF-IDF-weighted document vectors induced on the article body, obtaining article pairs likely reporting on the same event. The parameters of the TF-IDF vectors induction are the same as in Section 3.1. After aligning the candidate documents, article headings and the rest of the article text, referred as article body from now on, are processed separately due to different sampling strategies applied to these. We use a simple grid search and human judgment to establish the most promising region of similarity values in order to avoid candidate pairs with almost identical texts or candidates with similar topic but reporting on different events. While in news article bodies, we strive for balance between too low and too high similarity. In news headings, we target to select maximally dissimilar headings of news articles having maximally similar body texts as the most promising candidates for nontrivial paraphrase pairs. Furthermore, while the promising pairs of article body texts are selected for manual paraphrase extraction, news headings typically include only single sentence-like statements and are thus directly transferred into the paraphrase classification tool skipping the manual extraction phase. A total of approximately 2700 news heading pairs and 1500 article body pairs were used in the corpus construction.

### 3.3. Discussion forum messages

We hypothesize that different discussion forum messages related to same topics may include a sufficiently large number of naturally occurring paraphrases to justify a manual extraction effort. For example, different thread-starting messages under the same subforum often seek information on the same topic or share related experiences, or different replies to the same message often convey similar reactions. We set out to experiment with thread-opening messages with identical titles posted into the same subforum. We find that while most of the candidate document pairs selected this way are related messages from different authors often discussing similar personal experiences or seeking advice for similar matter. We also noticed a significant number of messages clearly written twice by the same user, with similar overall structure but using a different wording.

We use the public release of the Suomi24 discussion forum<sup>e</sup> including over 80M messages posted online between years 2001 and 2017. From the data release, we identify all thread opening messages and align candidate document pairs with identical title and subforum information combined with TF-IDF similarity of messages. Candidate alignments with too low or too high similarity, as well as candidates where the shorter message is merely a subset of the longer one, are filtered out based on preliminary human judgment gridding different similarity threshold values. This produced about 13K candidate message pairs. However, before the actual paraphrase extraction phase, 44% of these were yet discarded in an additional manual annotation step, where candidate document pairs were either accepted or rejected based on the potential estimated by inspecting the first few sentences from both documents. Here, the annotator only quickly verified a reasonable correspondence existing between the document pair without carefully reading the message content. This additional manual annotation step was carried out as we were not able to find an automatic method reliable enough to identify false positives among the candidates. Furthermore, filtering low-quality pairs before the actual paraphrase extraction step was found more efficient than executing filtering and paraphrase extraction simultaneously. In the end, a total of about 7100 accepted message pairs were used in the corpus construction.

### 3.4. Student translations

Seeing the potential of alternative translations originating from movie and TV episode subtitles, we initiated an attempt to find alternative source material where the same foreign text is translated into Finnish by multiple translators. One potential source of a constant stream of alternative translations is exercised from different language studies and courses, where several students translate the same exercise text. In order to avoid oversimplified short sentences, which one would see in many beginner level courses, we targeted exercises taken from university courses in translation studies where all students have sufficiently good skills and the exercises include translating authentic documents from different sources into Finnish. Such sources would typically include samples of magazine articles, business contracts, advertisements, etc.

We were able to collect 16 unique exercise texts with at least two different student translations. If more than two translations existed for the same source text, at most three different pairs were used in annotation so as to avoid over-extracting repetitive paraphrases, and a total of 28 document pairs were used in the corpus construction. However, the main limitation of student translations is their availability due to data usage regulations.<sup>f</sup>

<sup>e</sup><http://urn.fi/urn:nbn:fi:lb-2019021101>.

<sup>f</sup>Obtaining adequate permissions to use any student produced data involved manual permission inquiries and we found it difficult to motivate the students to give their consent. A long-term collaboration with a translation study program would likely improve this situation.

### 3.5. University exams

The final text source experimented with is student essays collected from university course exams, where the hypothesis is that all essays answering the same exam assignment will include similar arguments, and therefore, have a high probability for naturally occurring paraphrases. However, the student essays possess the same availability limitations as student translations where the usage of student materials is restricted and requires an explicit written consent.

We were able to collect a total of 34 student exams from three university courses (*Introduction to Language Technology*, *Corpus Linguistics and Language Technology*, and *Philosophy of Science and Research Process*). The exams included 24 unique questions or essay assignments for which at least one candidate pair (two alternative essay answers) was available. However, the answers for one assignment often divided into several subtopics because the students were able to select one aspect covered during the course to answer the assignment. The number of unique topics was consequently larger. We therefore processed each unique question/essay assignment/subtopic separately, rather than exams in full. The length of a typical answer varied between few sentences and one full page depending on the assignment. In the end, a total of 190 student answer pairs were used in the corpus construction.

## 4. Paraphrase candidate extraction

After the heuristic document alignment, the actual paraphrase candidate extraction is based on fully manual work. Next, we describe the paraphrase candidate extraction workflow, evaluate the adequacy of different text sources using several extraction measures, as well as show the distribution of paraphrases originating from different text sources in the final corpus.

### 4.1. Extraction workflow

Given a document pair extracted from one of the text sources, the manual annotation work begins with manual candidate extraction. In a dedicated candidate extraction tool, an annotator sees both documents simultaneously side-by-side and is instructed to extract all interesting paraphrases from the texts. In order to collect a varying set of nontrivial paraphrases, candidates with simple, uninteresting changes such as minor differences in inflection and word order are avoided during extraction. A paraphrase can be any text segment from few words to several sentences long, and the paraphrase extraction is not restricted to follow sentence boundaries. The two statements in one candidate pair can also be of different lengths, mapping for example one sentence on one side to several on the other side. The annotators are encouraged to select as long continuous statements as possible (rather than splitting them into several shorter ones), nevertheless at the same time avoiding over-extending one of the statements by including a long continuation which does not have a correspondence in its paraphrased version. The annotators are not actively trained to harmonize their personal candidate extraction strategies, since the aim is to include more diverse paraphrase candidates in the corpus, thus minor differences in extraction phase behavior are not considered harmful. The most typical property defining “personal style” in candidate selection was where to place the boundary between interesting and trivial pairs.

When completing the document pair, the annotator marks it finished and continues to the next document pair. After accumulating a reasonable amount of material in the extraction tool, all extracted paraphrase candidates are transferred into a separate paraphrase classification tool, where the annotation work continues as a separate session. Even if these two annotation phases were executed one after the other, the annotators were able to alternate freely between the two tasks in order to keep the working days more varied. Typically, the annotator who extracted the paraphrase candidates also did the labeling in the next phase. However, this is not strictly required and sometimes data is transferred between different annotators due to time constraints.

**Table 1.** Manual paraphrase extraction statistics for different text sources, where *Documents* refers to the number of document pairs producing paraphrases, *Empty* refers to the percentage of candidate document pairs not producing any paraphrase candidates (all other metrics are calculated after discarding the empty pairs), *Yield* refers to the average number of paraphrase pairs extracted from one document pair, *Coverage* is the total proportion of text (in terms of alphanumeric characters) selected in paraphrase extraction from the original source documents, and *Length* is the average length of the original document in terms of alphanumeric characters. Note that the alternative subtitle statistics are based on the first round of annotations only, where the movie/episode selection is not biased towards high-yield documents, and here one subtitling document refers to a 15-minute segment of a movie/episode

Text source	Documents	Empty (%)	Yield	Coverage (%)	Length
Alternative subtitles	2781	9.2	17.6	17.5	4300
News article bodies	1463	11.4	3.7	24.6	1600
Discussion forum messages	7106	36.7	1.7	22.8	500
Student translations	28	0.0	22.7	75.1	3700
University exams	190	31.6	2.4	25.8	1100

#### 4.2. Extraction statistics

Next, we analyze the different text sources used in the paraphrase extraction in several aspects. When evaluating the adequacy of the text source for the extraction purposes, we find it most interesting to measure how “productive” on average one document pair is. This is measured mainly using two metrics, the percentage of empty documents pairs, where empty refers to a document pair not producing any paraphrase candidates and can therefore be considered “useless” for the corpus construction purposes, as well as paraphrase yield, where yield refers to the average number of paraphrase candidates extracted from a nonempty document pair, where the assumption naturally is that the more one can extract from one document pair, the more time-efficient the extraction process is.

The overall extraction statistics are given in Table 1 separately for all five text sources. In terms of empty document pairs, the percentage varies between 0% and 37%, the two translation-based sources, student translations, and alternative subtitles, include the least amount of empty document pairs. An annotator not being able to extract any paraphrases from the document pair is typically caused by the two documents being lexically too similar and therefore not including interesting paraphrases, or them being topically related without any corresponding parts. In terms of the average yield of paraphrases per pair of documents, the story remains largely unchanged, with the two translation-based sources clearly having the best yield. From student translations, the annotators are able to extract on average 22.7 paraphrase candidates per nonempty document pair and from alternative subtitles the average yield is 17.6 candidates. In the end, it is not surprising that alternative translations yield the most amount of paraphrases as the translation process requires keeping the same basic information as present in the original, while for example in news articles the journalists can more freely select which aspects to report or not to report. Additionally, we were somewhat surprised how many verbatim quotations there were in news articles, where both news agencies clearly used the same reference text and possibly added a paragraph or two of their own text. The average length of the documents also naturally affects the yield, and the source with the worst average yield (discussion forum messages with only 1.7 paraphrase candidates per document pair) also has on average the shortest documents, with many of the discussion forum messages including only 1–2 sentences. In terms of coverage (proportion of the original text selected in paraphrase extraction), the differences are substantially smaller.

The final selection of source materials used for building the Turku Paraphrase Corpus is for the most part determined by two factors: availability and average paraphrase yield in the manual candidate extraction phase. Although the student produced materials were found promising in

**Table 2.** The number of paraphrase pairs in the released corpus originating from different text sources (rewrites, introduced in Section 5.3, are included in the statistics)

Text source	Paraphrase pairs	% of the corpus
Alternative subtitles	86,170	82%
News	9198	9%
<i>Body text</i>	5450	5%
<i>Headings</i>	3748	4%
Discussion forum messages	8175	8%
Student translations	760	1%
University exams	342	<1%

our experiments, especially the translation exercises which gave the best evaluation numbers in all metrics, the work required to settle legal restrictions on student produced materials prevented any larger-scale utilization of these sources under the scheduling constraints of the project. More groundwork would be required at the university and even national level to ease the usage of such data sources also retrospectively. Additionally, our goal of openly licensing (CC-BY-SA) the produced corpus creates increased complexity compared with a mere academic use in terms of student materials.

The limited amount of student materials left us with three primary text sources, of which alternative subtitles have a clearly better average yield per document pair compared with news articles and discussion forum messages. While news articles and discussion forum messages have better coverage (proportionally more of the source text is extracted), likely due to documents in general being shorter, one could assume the annotator being able to extract the same amount of material by just going through more document pairs. However, the amount of time the annotators spend on one document pair is considerably longer for news articles and discussion forum messages than for alternative subtitles. The main reason for this is that the two alternative subtitling documents are well aligned, while arguments in news articles and discussion forum messages often come in different order, requiring the annotators to scroll up and down in the paraphrase extraction interface in order to find the corresponding arguments. Also, after finding a corresponding argument in both documents, the annotator must yet verify the meaning of the extracted statement in the given context, as one cannot reliably assume the whole document following strictly the same story as in the case of the alternative translations where the source story is guaranteed to be identical. This extraction complexity effect is clearly visible in the weekly paraphrase extraction speed unofficially monitored throughout the project, where the extraction speed halved when switching from alternative subtitles to news articles and discussion forum messages. The extraction speed is thus the second limiting factor when selecting source material for annotation, and consequently, some of the text sources are highly overrepresented in the corpus. The number of paraphrase pairs obtained from different text sources are summarized in Table 2, the alternative subtitles dominating the final dataset with 82%, news texts and discussion forum messages both having a bit less than 10% portion, while both student materials represent only a tiny fraction of the corpus data.

## 5. Paraphrase annotation

After the candidate extraction, all candidate paraphrases are manually annotated according to the given annotation scheme. Next, we introduce this annotation scheme as well as some of the

more generally interesting annotation guidelines. In the end of the section, we present the overall annotation workflow where the annotators also have an option to provide an additional rewrite of the original paraphrase pair in order to correct small issues in the original candidates.

### 5.1. Annotation scheme

Many different paraphrase annotation schemes are presented in earlier studies, most commonly falling either into a simple yes/no (*equivalent* or *not equivalent*) as in MRPC (Dolan and Brockett 2005), or a numerical labeling capturing the strength/quality of paraphrases, such as the 1–4 scale (*bad*, *mostly bad*, *mostly good* and *good*) used in Opusparcus (Creutz 2018).

Instead of these simple annotation schemes, we set out to capture the level of paraphrasability in a more detailed fashion with an annotation scheme adapted to this purpose. Our annotation scheme uses the base scale 1–4 similar to many other paraphrase corpora, where labels 1 and 2 are used for negative candidates (unrelated/related but not a paraphrase), while labels 3 and above are paraphrases at least in the given context if not everywhere. In addition to base labels 1–4, the scheme is enriched with additional subcategories (flags) for distinguishing a small number of common special cases of paraphrases, which in many respects lie between the labels 4 (universal paraphrase) and 3 (paraphrase in the given context).

#### 5.1.1. Label 4: Universal paraphrases

Label 4 is assigned to cases of a universal (perfect) paraphrase that holds between the two statements in all reasonably imaginable contexts, meaning one can always be replaced with the other without changing the meaning. This ability to substitute one for the other in any context is the primary test for label 4 used in the annotation. Examples of universal paraphrases include:

Tulen puolella tunnissa.  
'I'll be there in half an hour.'  
Saavun 30 minuutin kuluessa.  
'I will arrive in 30 minutes.' → 4

Voin heittää sinut kotiin.  
'I can give you a lift home.'  
Pääset minun kyydissäni kotiin.  
'You can ride home with me.' → 4

Tyrmistyttävän lapsellista!  
'Shockingly childish!'  
Pöyristyttävän kypsytöntä!  
'Astoundingly immature!' → 4

With the base scale alone, a great number of candidate paraphrases would fail the substitution test for label 4 and be classified as label 3. This is especially true for any longer text segments which are less likely to express very strictly the same meaning even though conveying the same principal idea. So as to preserve some of the most important such general cases and to avoid overusing the label 3 category with a very diverse set of paraphrases, we introduce flags for finer subcategorization and therefore support a broader range of downstream applications of the corpus as well, since many applications may have different requirements for paraphrases. For instance, if considering rephrasing systems (paraphrase generation), the requirements for paraphrasing are quite strict in order to avoid for example the model learning to introduce additional facts or changing the style into offensive language on its own. On the other hand, in information retrieval,

the paraphrasing is usually more loosely defined, and finding occurrences with more variation is often appreciated. These annotated flags can only be attached to label 4 (subcategories of universal paraphrases), meaning the paraphrases are not fully interchangeable due to the specified reason, but, crucially, are context-independent that is their annotated relationship holds regardless of the textual context, which is unlike label 3. The possible flags are:

**Subsumption** (> or <). The subsumption flag is for cases where one of the statements is more detailed and the other more general (e.g. one mentioning *a woman* while the other *a person*), with the arrow pointing towards the more general statement. The relation of the pair is therefore directional, where the more detailed statement can be replaced with the more general one in all contexts, but not the other way around. The two common cases are one statement including additional minor details the other omits, and one statement being ambiguous while the other not. If there is a justification for crossing directionality (one statement being more detailed in one aspect while the other in another aspect), the pair falls into label 3 as the directional replacement test does not hold anymore. Examples of paraphrases with directional subsumption are shown below, where the first and second examples are cases of one of the statements including information the other omits (agent in the first example and purpose of the action in the second), while in the third example the latter statement is ambiguous, including both figurative and literal meaning:

Tulit juuri sopivasti.

'You arrived aptly.'

Loistava ajoitus.

'Fantastic timing.' → 4>

Tein lujasti töitä niiden rahojen eteen.

'I worked hard for that money.'

Paiskin kovasti töitä.

'I toiled away.' → 4>

En pysty tähän.

'I cannot do this.'

Tämä on liian suuri pala minulle.

'I'm in way over my head with this one.' → 4>

**Style** (s). The style flag is for marking tone or register difference in cases where the meaning of the two statements is the same, but the statements differ in tone or register such that in certain situations, they would not be interchangeable. For example, if one statement uses pejorative language or profanities, while the other is neutral, or one is clearly colloquial language while the other is formal. The style flag also includes differences in the level of politeness, uncertainty, and strength of the statements. Examples of paraphrases with different style (examples 1 and 2) and strength (example 3) include:

Helou gimmat!

'Hey, you gals!'

Päivää tytöt!

'Good day, girls!' → 4s

Mistä hitosta tietäisin?

'How the hell should I know?'

Minä en tiedä.

'I do not know.' → 4s

Täällä on aika kylmä ilmapiiri.  
 'The atmosphere is quite cold here.'  
 Täällä on jäätävä tunnelma.  
 'What a chilly mood there is round here.' → 4s

**Minor deviation (i).** The minor deviation flag marks in most cases minimal differences in meaning (typically *this* vs. *that*) as well as easily traceable differences in grammatical number, person, tense or such in cases where they are determined to have a difference in meaning. Some applications might consider these as label 4 for all practical purposes (e.g. information retrieval), while others should regard these as label 2 (e.g. automatic rephrasing). In cases where the minor change in for example mood or tense does not make a difference in meaning, the minor deviation flag is not marked. However, note that even when these minor differences are accepted, they cannot violate the paraphrasability in the context, for instance replacing the pronoun *minä* 'I' with *sinä* 'you' will not (generally speaking) make a paraphrase, while replacing *minä* 'I' with *me* 'we' can work in some contexts, however, quite rarely. Typical examples of paraphrases with minor deviation flag include:

Tämä laite on epäkunnossa.  
 'This piece of equipment is malfunctioning.'  
 Tuo kone on rikki.  
 'That machine is broken.' → 4i

Teitpä onnisti!  
 'You (plural) are in luck!'  
 Oletpa onnekas!  
 'Aren't you (singular) lucky!' → 4i

Vaimon mukaan hän vihaa tupakointia.  
 'According to his wife, he hates smoking.'  
 Hänen vaimonsa sanoo, että hän vihasi tupakan polttamista.  
 'His wife said that he hated smoking.' → 4i

The flags are independent of each other and can be combined in the annotation (naturally with the exception of > and < which are mutually exclusive).

### 5.1.2. Label 3: Context dependent paraphrases

Label 3 is a context dependent paraphrase, where the meaning of the two statements is the same in the present context, but not necessarily in other contexts. The common cases include statements, where both are ambiguous in different ways or both include different additional details not strictly necessary for conveying the main message (conflict in the subsumption flag directionality). Examples of context dependent paraphrases are shown below, where in the first example both include different additional details (first statement mentioning *night* while the second including a reference to *you*), while the second and third examples are cases where both statements are ambiguous in different ways or include a use case not covered by the other (e.g. in the third example the *911* can refer to the emergency number or simply be used when counting items, while the *emergency number* is *911* in some countries but not in all):

Miten eilisilta meni?  
 'How was last night?'  
 Miten teillä meni eilen?  
 'How did it go for you yesterday?' → 3



Aion tehdä kokeen.  
 'I am going to make an experiment.'  
 Aion testata sitä.  
 'I am going to test it.' → 3

911.  
 '911.'  
 Hätänumero.  
 'Emergency number.' → 3

### 5.1.3. Label 2: Related but not a paraphrase

Label 2 means related but not a paraphrase, where there is a clear relation between the two statements, yet they cannot be considered paraphrases in the sense outlined above for labels 4 and 3. Common cases include statements with a significant difference in the main message even if describing the same event, statements with contradictory information present, statements which could be paraphrases in some other context but not in their present context (such examples were very rare), or literal translations of metaphors which fail to communicate the metaphoric meaning in the source text (clumsy but understandable translations do receive label 3). Examples of related statements, which are not paraphrases are shown below, where the first example is topically heavily related and describing the same event but having a different main message, the second example describes the same event but from different point of time (therefore including contradictory information), and the third example includes a literal translation of a metaphor which doesn't make sense after the translation:

Tappion kokenut Väyrynen katosi Helsingin yöhön.  
 'After suffering defeat, Väyrynen disappeared into the night of Helsinki.'  
 Väyrynen putoamassa eduskunnasta.  
 'Väyrynen is in danger of dropping out of the Finnish Parliament.' → 2

Aurassa perjantaina kadonnut 12-vuotias poika löytynyt.  
 'The 12-year-old boy who went missing in Aura on Friday has been found.'  
 Poliisi etsii 12-vuotiasta poikaa Aurassa.  
 'The police are searching for a 12-year-old boy in Aura.' → 2

Olet löytänyt onnen.  
 'You have found happiness.'  
 Nyt sinulla on avaimet linnaan.  
 'Now you have the keys to the castle.' → 2

### 5.1.4. Label 1: Unrelated

Label 1 is for unrelated candidates, where there is no reasonable relation between the two statements, most likely occurring due to a false positive in candidate selection. If the candidate pair shares only a single proper name while the topic otherwise is different, the candidate is considered unrelated.

Oletteko Sherlock Holmes?  
 'Are you Sherlock Holmes?'  
 Riippuu.  
 'It depends.' → 1

Sipoonranta on Sipoossa, ei Helsingissä.  
 'Sipoonranta is located in Sipoo, not in Helsinki.'  
 Sipoonranta hakee taas lisääaikaa rakentamiseen.  
 'Sipoonranta is again applying for more time for building.' → 1

#### 5.1.5. Label x: Skip

If labeling a candidate pair is not possible for another reason, or giving a label would not serve the desired purpose (e.g. wrong language or identical statements), the example can be skipped with the label x.

## 5.2. Annotation guidelines

While each decision in paraphrase annotation must be done based on considering each individual example separately, several systematic differences among the annotators were identified during the annotation process, and comprehensive annotation guidelines were produced to guide the annotation process towards harmonized decisions between different annotators. A total of 17-page annotation manual was produced in collaboration among the annotators, and the guidelines were revised and extended regularly to account for new problematic cases. The full manual is published as a technical report (Kanerva *et al.* 2021a), and some of the most interesting/relevant policies are discussed below.

### 5.2.1. Syntactic structure

Merely syntactic differences are not accounted in the labeling if they do not change the sentence meaning, even if the difference would make sentence substitution clumsy in some contexts. For example, the lack or inclusion of discourse connectives can make the sentence feel clumsy or isolated from the context, however they barely carry much additional information. The same policy is adapted to for example differing verb tense and mood if the difference does not carry change in meaning. However, if a shift in meaning is noticed it is annotated accordingly.

### 5.2.2. World knowledge

In certain cases, one of the statements includes additional information which can be seen as world knowledge (facts generally known or knowable by everyone). For example, in the paraphrase pair

Omena on hedelmä, josta valmistetaan mm. hilloa ja mehua.  
 'An apple is a fruit from which you make jam and juice, among other things.'  
 Omenasta valmistetaan muun muassa hilloa ja mehua.  
 'Among other things, jam and juice are made from apples.'

the second statement does not explicitly mention apple being a fruit. However, considering that this is a generally acknowledged fact, which does not contribute to the core meaning, explicitly mentioned additional world knowledge facts are not considered additional information in paraphrase annotation, and therefore, the above-mentioned example would receive label 4 in annotation.

The same principle is adapted for well-known noun modifiers (e.g. permanent titles and descriptive nouns such as *Queen Elizabeth II*, *ski jumping legend Matti Nykänen* or *tech company Microsoft*). However, if the noun modifier is considered to be meant for temporary use only, as many times for example in politics (e.g. *prime minister Sanna Marin*), noun modifiers are considered additional information as it binds the statement into a specific time.

In few cases, the world knowledge principle allows proper name replacement with a common noun phrase, if the entity can be unambiguously individualized from the common noun description. For example, in the paraphrase pair

Ensimmäinen avaruuteen lähetetty suomalaissatelliitti tuhoutui.  
 'The first Finnish satellite that was launched to space was destroyed.'  
 Aalto-2 tuhoutui.  
 'Aalto-2 was destroyed.'

while *The Finnish satellite* could refer to any Finnish satellite, there can be only one “first one”, which then individualizes the noun phrase and the example is annotated with label 4.

### 5.2.3. Time references

Time references can be either exact (*24.12.1999, in 2020, 16:00 o'clock*) or relative with respect to the current time (*today, last year, in three hours*). When comparing two exact time expressions, the label is 4 if the same amount of information (e.g. day, month, year) is present, but often 4 with a subsumption flag if one of the two is more descriptive and the additional information cannot be considered world knowledge. When comparing two different relative time references with each other (e.g. *in the beginning of the week* and *three days ago*), the label is usually 3 if the time is not further specified elsewhere in the statements. When comparing exact time with relative time, the labels depends on whether the exact time can be considered world knowledge or not. For example, in statements

Matti Nykänen kuoli viime vuoden helmikuussa 55-vuotiaana.  
 'Matti Nykänen died in February of last year at the age of 55.'  
 Matti Nykänen kuoli helmikuussa 2019. Hän oli kuollessaan 55-vuotias.  
 'Matti Nykänen died in February of 2019. He was 55 years old at the time of his death.'

the date of death of a famous person can be considered world knowledge, and the paraphrases can be labeled with label 4> the latter being more general as it can be used in any point of time, while *February of last year* can only refer to the year 2019 in this context and therefore be used only in 2020. When comparing exact time with relative time in the context of events not considered world knowledge, for example in

Rikos tapahtui viime vuoden helmikuussa.  
 'The crime happened in February of last year.'  
 Rikos sattui helmikuussa 2019.  
 'The crime took place in February of 2019.'

the event in question (crime) is not individualized and the exact time cannot be considered world knowledge, therefore the label is 3.

### 5.3. Annotation workflow

After accumulating a reasonable amount of material in the candidate extraction phase (typically every two to three days), the extracted paraphrase candidates are transferred into a dedicated paraphrase classification tool, where the annotator is able to see all paraphrases extracted from the document pair one by one. In the paraphrase classification tool, the annotator assigns a label for each paraphrase candidate using the above-mentioned annotation scheme. Even if the extracted paraphrases are shown one-by-one in the tool, the full document context is available. In addition

to labeling, the tool provides an option for rewriting the paraphrase pair to be fully interchangeable, universal paraphrases. The annotators are instructed to rewrite paraphrase pairs that are not already label 4, in cases where a simple edit, for example word or phrase deletion, addition, or re-placement with a synonym or changing an inflection, can be easily constructed. Rewrites must be such that the annotated label for the rewritten example is always label 4. In cases where the rewrite would require more complicated changes or would take too much time, the annotators are instructed to move on to the next candidate pair rather than spend time on considering the possible rewrite options.

The classification tool also provides an option to tag examples where the annotator feels unsure about the correct label, the example is particularly difficult, or otherwise more broadly interesting. These examples were discussed in the whole annotation team during daily annotation meetings. The annotators also communicated online, for instance seeking a quick validation for a particular decision.

#### **5.4. Ensuring annotation consistency in early annotations**

As the annotation guidelines were revised and extended throughout the corpus annotation, there is the potential of small discrepancies between examples annotated at the very early stage of the project compared with those annotated at the very end. In order to assure the consistency between the revised guidelines and early stage annotations, during the final weeks of annotation several quality assurance rounds were carried out, especially targeting labels whose guidelines changed during early annotation work.

All annotated examples were first divided by labels, and then sorted based on annotation timestamps from earliest to latest. Concentrating on the most problematic labels *s* (flag for style) and *i* (flag for minor deviation), examples including these flags were manually checked and corrected if necessary, starting from the earliest annotations and continuing until the latest guidelines and the annotated examples were in sync, and no systematic errors were noticed anymore. A total of 5.7% of all annotated examples were inspected, of which about 30% were corrected according to the latest guidelines. Time-wise most corrections were dated to the first 2 months of the annotation work.

## **6. Corpus statistics and evaluation**

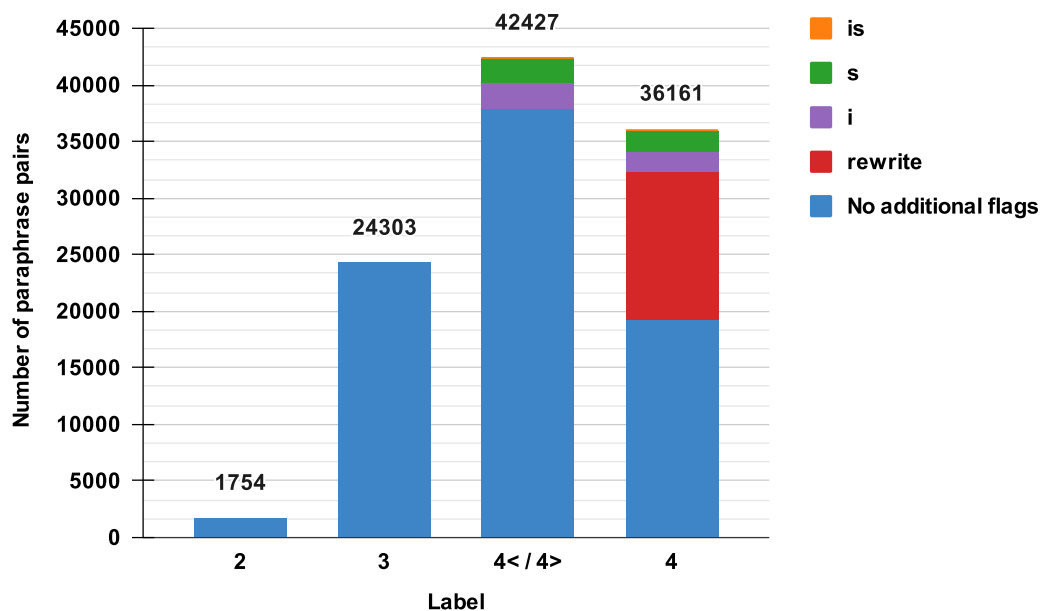
The released corpus is comprised of 91,604 naturally occurring paraphrase pairs extracted from the source documents with an additional 13,041 rewrites, thus resulting in a total of 104,645 manually classified Finnish paraphrase pairs. The data are randomly divided into training, development, and test sections using a 80/10/10 split; however, with the restriction that all paraphrases from the same movie, TV episode, news article, student translation text, or exam question are assigned to the same section. Basic statistics are summarized in Table 3, and the label distribution is shown in Figure 1. As the manual candidate extraction targeted “true” paraphrases, 98% of all annotated paraphrases are classified to be at least paraphrases in their given context (label 3) if not in all contexts (label 4). The number of candidates labeled with labels 1 or *x* is negligible, therefore these are discarded from the corpus release altogether.

### **6.1. Annotation quality**

The annotation work was carried out by six main annotators together with a broader project team supporting their effort. The six annotators used a total of 30 person-months for the corpus construction, where the work includes paraphrase extraction, label annotation as well as other related tasks such as guideline documentation. Each annotator had a strong background

**Table 3.** The sections of the corpus and their sizes in terms of number of paraphrase pairs

Section	Examples	Rewrites	Total
Train	73,165	10,480	83,645
Devel	9231	1298	10,529
Test	9208	1263	10,471
Total	91,604	13,041	104,645

**Figure 1.** Label distribution in the whole corpus.

in language studies with an academic degree or ongoing studies in a field related to languages or linguistics. After the initial training phase, most of the annotation work was carried out as single annotation. However, in order to monitor annotation consistency, double annotation batches were assigned regularly. In double annotation, one annotator first extracted the candidate paraphrases from the aligned documents, but later on these candidates were assigned to two different annotators, who annotated the labels independently from each other. Afterwards, the two individual annotations were merged and conflicting labels resolved together with the whole annotation team. These consensus annotations constitute a consolidated subset of the data, which can be used to evaluate the overall annotation quality by measuring individual annotators against this subset.

A total of 2025 examples (2% of the paraphrases in the corpus, excluding rewrites) were double annotated, most of these being annotated by exactly two annotators; however, some examples may include annotations from more than two annotators, and thus the total amount of individual annotations for which the consensus label exists is bit more than twice the number of double annotated examples (4287 annotations in total). We measure the agreement of individually annotated examples against the consolidated consensus annotations in terms of accuracy, that is the proportion of individually annotated examples where the label matches the consensus annotation.

The overall accuracy is 70% when using the full annotation scheme (base labels 1–4 as well as all flags). When discarding the least common flags *s* and *i* and evaluating only base labels and directional subsumption flags, the overall accuracy is 74%.

In addition to agreement accuracy, we calculate two versions of Cohen's kappa, a metric for inter-annotator agreement taking into account the possibility of agreement occurring by chance. First we measure the kappa agreement of all individual annotations against the consolidated consensus annotations, an approach typical in paraphrase literature. This kappa is 0.63, indicating substantial agreement. Additionally, we measure the Cohen's kappa between each pair of annotators. The weighted average kappa over all annotator pairs is 0.42 indicating moderate agreement. Both are measured on full labels. When evaluating only on base labels and directional subsumption flags, these kappa scores are 0.66 and 0.45, respectively.

Direct comparison of annotation agreement with other manually annotated paraphrase corpora is not straightforward due to several factors affecting the expected agreement measures, the most influential factors likely being the labeling scheme and label distribution of the corpora. While the kappa measure tries to account for this, this is especially true for accuracy. It should also be noted that in many semantic annotation tasks, agreement scores can only be used as estimates, and low score does not necessarily refer to a low annotation quality, but rather the nature of the task itself. (Pavlick and Kwiatkowski (2019), Davani, Díaz, and Prabhakaran (2022)) When comparing to other paraphrasing projects, all our metrics are in the same ballpark with other manually annotated samples, MRPC reporting accuracy of 84% with binary labels, Opusparcus accuracy between 64% and 67% with four labels, and ParaSCI reporting kappa of 0.71 when measuring the individual annotator against the majority vote on a five label scheme. Furthermore, one must also note that while our manual annotation primarily focuses on distinguishing between different positive labels, the other annotation efforts mentioned include also substantial amount of negatives, making the task slightly different from ours.

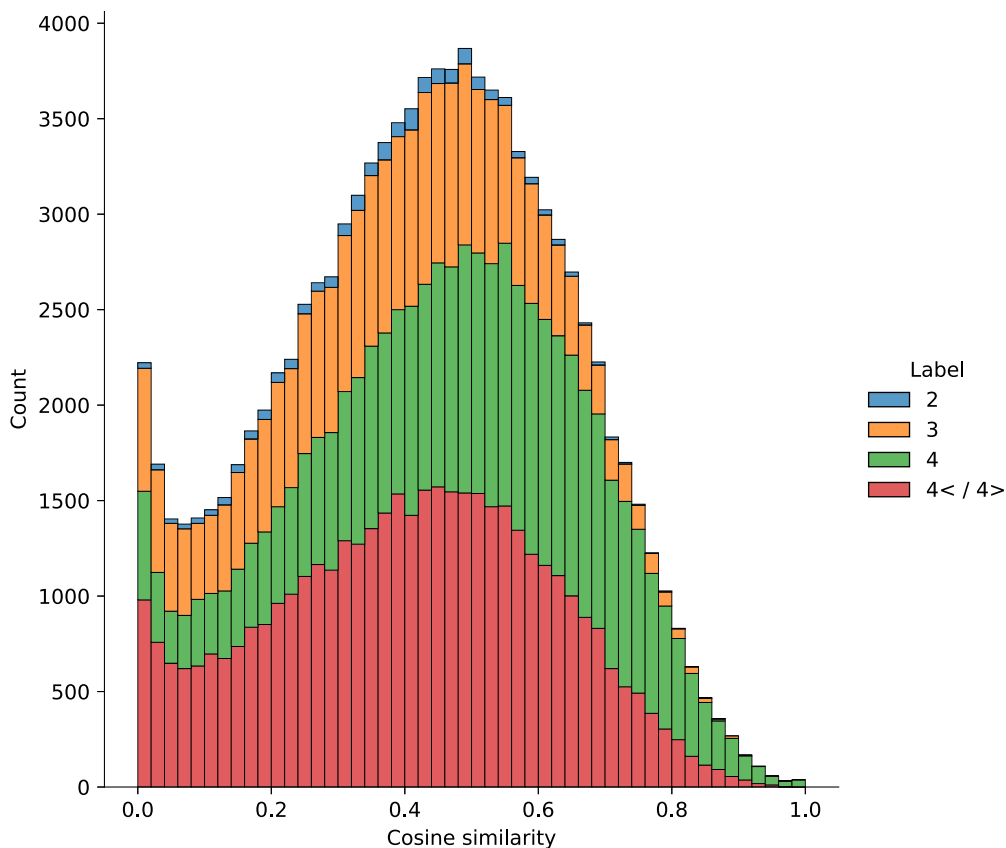
### 6.1.1. Rewrites

As mentioned earlier, during the annotation, the annotators have the possibility to rewrite the statements if the classification is anything else than pure label 4. This can be interpreted as the annotators fixing all flaws in the paraphrases and turning the candidates into perfect, context independent paraphrases. In order to evaluate the assumption of the rewrites always being a pure label 4, we sample 500 rewrites for double annotation. To ensure that the annotator does not know whether the candidate is a rewrite or normal extracted paraphrase, the rewrites are mixed together with normal paraphrase candidates in a 50/50 ratio. In addition, during this experiment, the document context is hidden in the annotation tool, as the context has a potential to reveal the candidate being a rewrite. The data are distributed in a fashion where all annotators receive only candidates previously annotated by someone else so that there is no risk of the annotators recalling the previously annotated examples. The candidates are also randomly shuffled.

After merging and resolving the double annotated examples, 78% of rewrites received the label 4. This is on par with the overall annotation consistency, showing the quality of rewrites largely following that of the natural examples in the corpus.

## 6.2. Lexical diversity and corpus comparison

One of our main goals was to obtain a set of paraphrase examples that are not highly lexically similar. In Figure 2, we measure the distribution of different labels in the corpus conditioned on the cosine similarity of the paraphrase pairs calculated using TF-IDF weighted character n-grams of lengths 2–4. While the different positive labels are evenly distributed in the low lexical similarity area up until similarity value 0.5, in the high similarity area the label 4 begins to dominate the data.



**Figure 2.** Histogram of different labels in the corpus conditioned on cosine similarity of the paraphrase pairs.

However, as can be seen from the figure, most of the paraphrases in the corpus fall into the low or mid-range similarity area making the high similarity quite sparsely populated.

Next, we compare our corpus with the two existing Finnish paraphrase candidate corpora, Opusparcus and TaPaCo using three different metrics: (1) the distribution of the lengths of the paraphrased segments, (2) the distribution of lexical similarity values of the two paraphrased statements, and (3) the presence of systematic paraphrasing patterns that can be identified automatically.

Such direct comparison between different corpora is naturally complicated by several factors. Firstly, compared with our manually annotated paraphrases with significant bias towards positive labels, both Opusparcus and TaPaCo consist primarily of automatically extracted paraphrase candidates, and the true label distributions are mostly unknown. The small manually annotated development and test sections of Opusparcus are sampled to emphasize lexically dissimilar pairs, and therefore not representative of the characteristics of the rest of the corpus, limiting their usage for corpus comparison purposes. We therefore compare with the fully automatically extracted sections of both Opusparcus and TaPaCo, as these represent the bulk of the corpora. In our corpus, we can discard the small proportion of examples of label 2, that is the examples known to not be paraphrases, while the automatically extracted sections of Opusparcus and TaPaCo are expected to include a significant portion of negative paraphrase examples as well. Therefore, when drawing any conclusions an important factor to consider is that the characteristics of false and true candidates may differ substantially, false candidates for example likely being on average more dissimilar in terms of lexical overlap than true candidates.

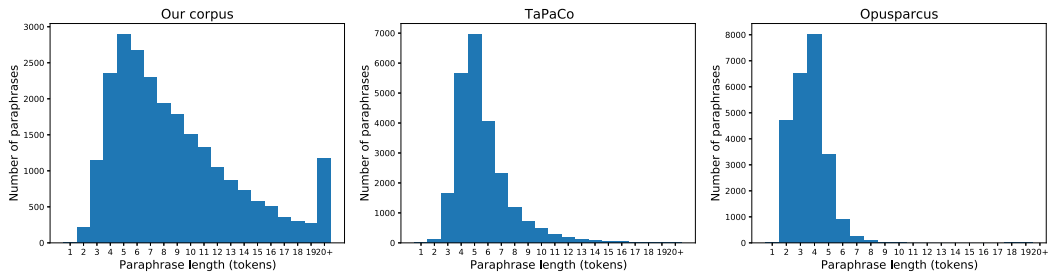


Figure 3. Comparison of paraphrase length distributions in terms of tokens per paraphrase.

For each corpus, we sample 12,000 paraphrase pairs in order to keep the sizes of the compared sets uniform. For our corpus, we selected a random sample of true paraphrases from the train section. For TaPaCo, the sample covers all paraphrase candidates from the corpus, however with the restriction of taking only one, random pair from each ‘set’ of paraphrases, while for Opusparcus, which is sorted by a confidence score in descending order, the sample was selected to contain the most confident 12K paraphrase candidates.<sup>g</sup>

From Figure 3, it can be seen that the distribution of the paraphrase lengths in our corpus is wider and contains a hatrivial amount of longer paraphrases as well, while the other two corpora mainly contain relatively short paraphrase candidates. The average number of tokens in our corpus is 8.8 tokens per one paraphrase statement, while it is 5.6 in TaPaCo and 3.6 in Opusparcus. Furthermore, as the manual paraphrase extraction was not tied to follow sentence boundaries in our corpus, we measure how many of our paraphrases are short phrases, single sentences, or longer than a sentence. To this end, we apply a Finnish dependency parser (Kanerva *et al.* 2018) to segment sentence boundaries and recognize whether a sentence is well-formed (starts with a capitalized letter, ends with a punctuation character and includes a main verb) or not. We find that approximately 12% of the paraphrase statements are phrases or not well-formed single sentences, 73% are well-formed, single sentences, 13% are two sentences long, and the remaining 2% being segments which are more than two sentences long. When looking into paraphrase pairs instead of individual paraphrase statements, 63% of the pairs have one-to-one mapping of well-formed sentences, following with one-to-two (10%), sentence-to-phrase (9%), phrase-to-phrase (7%), and two-to-two (7%) mappings, the other variants occurring only rarely.

Figure 4, the cosine similarity distribution of the paraphrase pairs is measured using TF-IDF weighted character n-grams of length 2–4 for these three corpora. This allows us to establish to what degree the corpora contain highly lexically distinct pairs. From this figure, it can be seen that our corpus has a larger proportion of paraphrases with lower lexical similarity, while the distribution of the other two corpora are skewed towards pairs with higher lexical overlap.

Finally, we study the corpora from the point of view of systematic paraphrasing patterns, that is pairs which are formed in a systematic, predictable manner. To this end, we follow the method used in our prior work (Chang *et al.* 2021), recognizing six systematic ways in which the two segments of a paraphrase pair differ from each other: (1) word reordering, (2) word inflections (both having same lemmas in the same order), (3) lemma reordering, (4) lemma reordering after excluding all functional words (both having the same content word lemmas), (5) synonym replacements, and (6) a combination of (4) and (5).<sup>h</sup> These six types of differences are automatically detectable

<sup>g</sup>When the length analysis was repeated with a sample of 480K most confident pairs, the length distribution and average length remained largely unchanged, while the similarity distribution became close to flat. Without manual annotation, it is hard to tell the reason for this behavior.

<sup>h</sup>If a paraphrase pair can be accounted by either disregarding functional words or synonym substitution, it is classified as disregarding functional words.



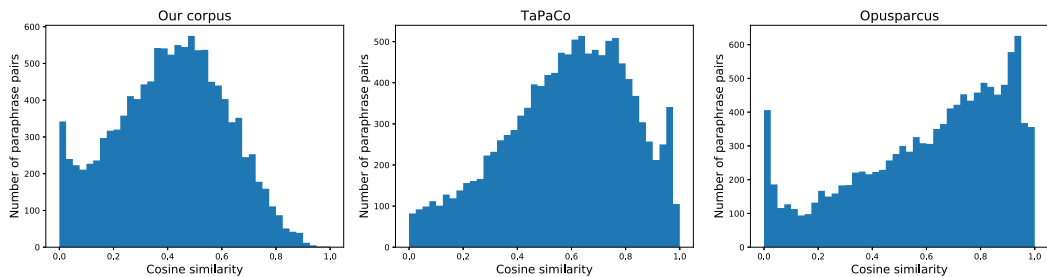


Figure 4. Comparison of paraphrase pair cosine similarity distributions.

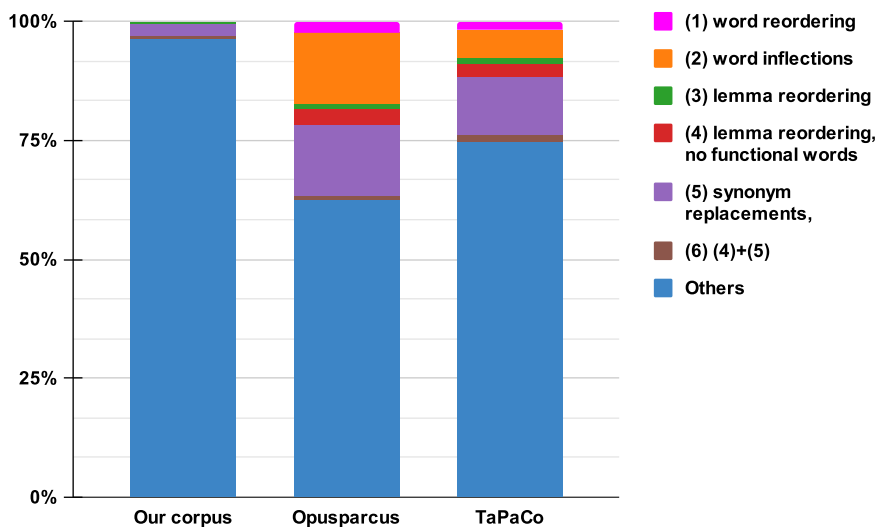


Figure 5. Percentage of the types of systematic differences characterizing the paraphrases in Opusparcus, TaPaCo, and our corpus. *Others* refers to all paraphrases including differences not automatically detectable by the used method.

with a simple approach and can be therefore regarded as to some degree “trivial” paraphrase pairs. From Figure 5, it can be seen that our corpus has a notably smaller proportion of trivial paraphrases than Opusparcus and TaPaCo. While the other two corpora have a larger proportion of paraphrases that can be accounted for by lemmatization, that is type (1), (2), and (3), our corpus has less than 1% of these each. The most prominent type of trivial paraphrases in our corpus is synonym replacement, at 2%. These results support that our manually extracted paraphrases contain more interesting, nontrivial paraphrases than automatically collected corpora and help to validate our manual extraction approach.

## 7. Paraphrase classification

Having described the paraphrase corpus itself, we will continue to paraphrase modeling experiments. We first apply a pairwise paraphrase classifier, where for a given candidate pair the classifier predicts the label based on the labeling scheme used in the corpus. While the classification model could be straightforwardly trained using only the annotated paraphrase corpus, in addition to such a baseline model we also apply a bootstrapping approach where the training data is augmented with automatically extracted negative pairs to account for the low frequency of negative pairs in the original corpus.

When creating the paraphrase corpus, we concentrated on building a dataset of nontrivial paraphrases classified as positive in manual annotation (label 3 and above), where the occasional label 2 paraphrase candidates were only a by-product of the annotation work. However, in order to train models able to distinguish negative candidates from the positives, a sufficient number of negative examples is required during the model training. While unrelated negative candidates (label 1) can be obtained trivially by pairing arbitrary sentences, it is shown for example by Guo *et al.* (2018) in the context of parallel data mining that it is not sufficient to introduce negatives based only on arbitrary pairs. Instead, better results can be obtained by including hard negatives, that is candidates which share for example topic or are otherwise related while still not being paraphrases.

In order to obtain such training data for the paraphrase classifier, in our bootstrapping approach we use sentence embeddings obtained from a basic language model without task-specific fine-tuning to select semantically related pairs of sentences from a large corpus of text. These are subsequently filtered using an initial classifier trained purely on the manually annotated corpus data, preserving examples with a confident negative prediction. Finally, we train new models for paraphrase classification using a combination of the manually annotated corpus and the automatically extracted negative candidates. Next, we describe all these steps in detail.

### 7.1. Paraphrase classifier

Our paraphrase classification model is a pairwise classifier based on the BERT encoder, following our initial work reported in Kanerva *et al.* (2021b). The model receives one candidate pair at a time, encoded as the sequence [CLS] A [SEP] B [SEP], where A and B are the two paraphrase statements and [CLS] and [SEP] the special tokens in the BERT model. The classifier is a multi-output model implemented on top of the pretrained FinBERT language model (Virtanen *et al.* 2019), including four separate prediction layers, one for the base label (with classes 2, 3, or 4), one for the subsumption flag (<, > or none), one for the style flag (s or none), and one for the minor deviation flag (i or none). As the additional flags only apply to examples where the base label is 4, no gradients are produced for subsumption, style, and minor deviation prediction layers if the base label of the example is 2 or 3. The predictions are based on five different embeddings obtained from the final BERT layer: embeddings for the [CLS] and the two [SEP] tokens, as well as the average of token embeddings calculated separately for statement A and statement B, all five concatenated together and projected for the four prediction layers. The overall model design (e.g. concatenating the five embeddings rather than using the plain [CLS] embedding) is optimized during preliminary experiments conducted on the development data. The use of multiple output layers rather than treating each label combination a separate class in standard multiclass classification is chosen to account for certain flag combinations, such as 4>is, which would not be predicted at all by a standard multiclass model as such label combinations are so rare in the data.

The initial classifier is trained on the Turku Paraphrase Corpus using the data split reported in Table 3, receiving an accuracy of 58.1 and a weighted average F-score of 57.6 when tested on the corpus test set treating each complete label as its own class during evaluation. As expected, the initial classifier is weakest at classifying the small amount of negative examples (label 2) in the test set, giving an F-score of 30.3 for label 2, and fully reflecting the design choices of the corpus. The full evaluation numbers for the initial classifier are given later in Section 7.3 (Table 4) where the results are compared with the final, bootstrapped model.

### 7.2. Extracting candidate pairs for model bootstrapping

Deep language models, such as BERT (Devlin *et al.* 2019) or LASER (Schwenk and Douze 2017), are commonly used as general sentence encoding methods, assigning dense vector representations to sentences and other short text segments. Simple metrics, such as cosine similarity or Euclidean distance, can then be used to efficiently estimate the similarity of two sentences in the vector space,

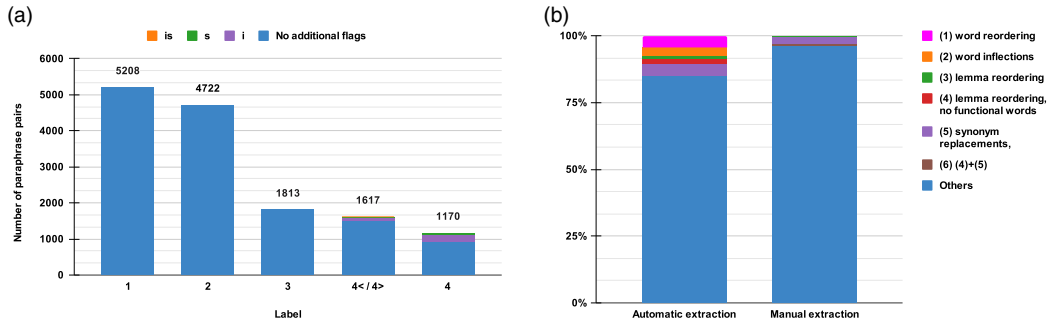
**Table 4.** Baseline classification performance on the two test sets, when the base label and the flags are predicted separately. In the upper section, we merge the subsumption flags with the base class prediction, but leave the flags *i* and *s* separated. The rows *W. avg* and *Acc* on the other hand refer to performance on the complete labels, comprising all allowed combinations of base label and flags. *W. avg* is the average of P/R/F values across the classes, weighted by class support. *Acc* is the accuracy

Turku Paraphrase Corpus test set					Opus-parsebank-test				
Label	Prec	Rec	F	Support	Label	Prec	Rec	F	Support
2	46.8	22.4	30.3	161	neg	99.0	23.1	37.5	6712
3	60.3	50.9	55.3	2434	3	11.7	48.3	18.8	1146
4<	55.8	57.9	56.8	2003	4<	36.9	64.7	47.0	425
4>	57.0	61.9	59.4	2287	4>	37.8	70.7	49.3	560
4	70.5	74.3	72.4	3586	4	47.1	91.3	62.1	793
<i>i</i>	50.0	47.4	48.6	454	<i>i</i>	52.0	71.3	60.2	164
<i>s</i>	49.1	37.0	42.2	438	<i>s</i>	28.2	48.0	35.6	50
<i>W. avg</i>	57.7	58.1	57.6		<i>W. avg</i>	77.8	35.5	37.9	
<i>Acc</i>			58.1		<i>Acc</i>			35.5	

with sentences equivalent or closely related in meaning being also highly similar in terms of these metrics. We rely on such embedding similarities in order to find promising, initial candidates of related sentences for model bootstrapping, where our aim is to collect negative pairs including a nontrivial topical overlap (hard negatives). For creating the sentence embeddings, we use the vanilla BERT model pretrained for Finnish without any task specific fine-tuning of the model.

In order to obtain enough candidate sentences for collecting hard negatives for our bootstrapping experiments, we use two different data sources: OPUS and Finnish Internet Parsebank. OPUS (Tiedemann 2012) is an open parallel corpus collecting a diverse set of parallel sentences ranging from EU legislation and software manuals to movie subtitles. The OPUS data is obtained through the data release of the Tatoeba translation challenge (Tiedemann 2020). The Finnish Internet Parsebank (Luotolahti *et al.* 2015) is a large-scale Finnish corpus collected through dedicated web crawls targeted to find high quality Finnish material from the Internet. Together, these two resources include almost 400M unique sentences. All unique sentences in this collection are first encoded with the FinBERT model of Virtanen *et al.* (2019) taking the average of token embeddings to obtain one vector for each sentence. Next, for each sentence, its five most similar sentences are collected from the same source (OPUS or Parsebank) using Euclidean distance of the embeddings implemented in the FAISS library (Johnson, Douze, and Jégou 2021) for fast similarity comparison, constituting a massive candidate set of  $400M \times 5$  closely related sentence pairs. Finally, all duplicate pairs (irrespective of direction) are discarded.

To understand the distribution of different paraphrase labels in this set of candidates, we selected a random sample for manual annotation. A total of 15,000 sentence pairs are sampled, taking 7500 pairs from both OPUS and Finnish Internet Parsebank. So as to maximize the informativeness of this manual evaluation, we stratify the sample in terms of lexical similarity, measured as cosine similarity of term frequency (TF) vectors based on character *n*-grams of lengths 2–4. All candidate pairs are split into 20 lexical similarity intervals in increments of 0.05, with an equal number of pairs selected from each interval for manual annotation. This stratified sampling together with manual annotation allows us to estimate the distribution of different labels in each similarity interval separately. In the manual annotation, we labeled 14,530 candidate pairs (470 were skipped with label *x* during the annotation due to various issues such as incorrect

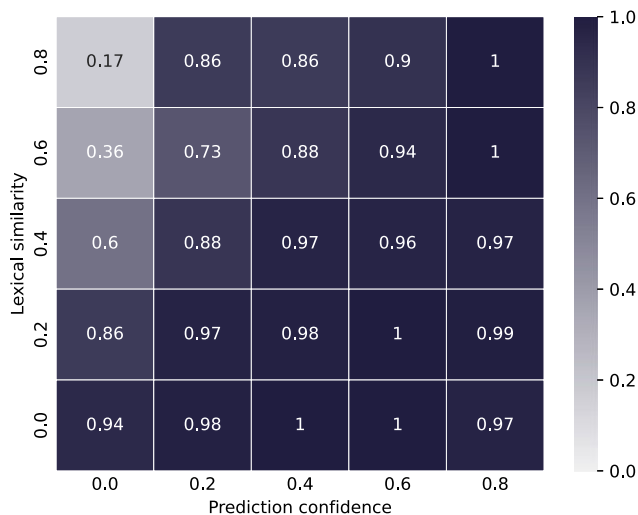


**Figure 6.** (a) Distribution of the manually annotated labels in the opus-parsebank set including both development and test examples. (b) Comparison of the types of paraphrases in the manually and automatically extracted data. The *manually-extracted data* refers to the training set of our corpus, while the *automatically extracted data* refers to the combination of opus-parsebank-dev and opus-parsebank-test sets.

language or whitespace-only differences). The sample is divided into development and test sections (hereafter opus-parsebank-dev and opus-parsebank-test), with a 1/3 and 2/3 split. While the development section is used to analyze the different properties of the data, the test section is reserved only for the final test purposes, and none of the annotated data is used for the actual model training.

Next, we analyze the annotated sample from several perspectives. In Figure 6, on the left we show the label distribution of this sample, and on the right side we plot the automatically detectable systematic paraphrasing patterns introduced in Section 6.2. Contrary to the manually extracted corpus, the sample does not strive to exclude uninteresting candidates including only elementary variation, and among the examples with a high lexical similarity, trivial differences are included, such as differences purely in punctuation or capitalization. While the manually constructed corpus included only occasional negative paraphrases, as expected, the label distribution in the opus-parsebank sample is skewed towards negative paraphrases (68% being annotated with label 1 or 2). When measuring the automatically detectable systematic paraphrasing patterns among positive examples (labels 3 and above), the figure confirms the higher tendency towards trivial variation appearing among the automatically extracted paraphrases than among the manually selected ones. Along with those shown in the figure, additional 2% of positive paraphrase pairs in the opus-parsebank sample contain only small character differences that are usually typos or punctuation differences, totaling the recognized elementary variation to cover approximately 17% of all positive paraphrase pairs. However, part of the elementary variation can be explained by the stratified sampling over lexical similarity values, as high similarity areas have proportionally more elementary variations, compared with the automatically detected paraphrase candidates with lower similarity ranges, which are mostly negative paraphrase pairs, along with some nonelementary positive paraphrase pairs.

Finally, we analyze the annotated sample regarding the reliability of the classifier prediction scores, with the aim of identifying areas where we can be reasonably confident in the classifier predictions and sample “safe” negative examples to complement the primary manually annotated corpus. When simultaneously plotting classifier prediction scores (probability of negative label) together with the lexical similarity intervals into a two-dimensional plot, we are able to divide the examples into several tiles, which can furthermore be enriched with the manually annotated labels to estimate the actual amount of negative candidates (labels 1 and 2) in each tile. This information can be used to select tiles (and their corresponding lexical similarity and prediction score values) to collect safely negative or safely positive paraphrase candidates when applying the same metrics for the full collection of closely related pairs. The observed tiles are demonstrated in Figure 7,



**Figure 7.** Heatmap with estimated negative example density per tile in increments of 0.2 for opus-parsebank-dev. Lexical similarity is plotted in y-axis and prediction confidence in x-axis, creating two-dimensional tiles when both are divided in increments of 0.2. Each tile is yet enhanced with a density score indicating the percentage of negative examples in the tile based on the manually annotated labels.

where the data is split into five intervals in increments of 0.2 on both axes, as both the prediction score and lexical similarity values range between 0 (unsure, highly dissimilar) and 1 (confident, highly similar). Each tile is yet enhanced with an annotation indicating the percentage of negative labels in the tile estimated using the manually annotated sample.

For collecting negative candidates, all pairs with lexical similarity of under 0.1 or negative class prediction confidence over 0.4 were chosen as optimal region. When applying these values across the whole set of closely related sentence pairs (discarding those in the annotated sample), we were able to extract approximately 5M nonparaphrase candidates with precision of 97.7% as estimated from the manually annotated sample. Additionally, the same experiment was repeated for the positive paraphrase candidates by using lexical similarity of over 0.5 and the model’s prediction confidence score of 0.998 or greater for the base label 4, obtaining a set of 500K positive paraphrase candidates with estimated precision of 95.8%. Both datasets are released as supplementary data together with the manually annotated examples in order to support for example training with a binary objective (paraphrase or not-a-paraphrase).

### 7.3. Paraphrase classification results

For the final classification experiments, the manually annotated Turku Paraphrase Corpus training set of 84K pairs is combined with an additional 84K pairs sampled from the automatically gathered negatives, therefore creating a somewhat balanced set of positive and negative training examples. While all manually annotated examples naturally include the full label information, for automatically gathered “training” negatives, we do not have distinction between the two negative labels (label 1 and label 2), and therefore we opted to use only a single label for all negative examples while training the classifier. As shown in Table 5 versus the baseline performance shown in Table 4, besides slightly improving the label 2 classification performance, enhancing the training data with automatically gathered negatives does not affect the performance on the Turku Paraphrase Corpus test set, where the great majority of the test examples fall into the different positive labels. Therefore, the automatically gathered negative training examples do not

**Table 5.** Final classification performance on the two test sets, as in Table 4

Turku Paraphrase Corpus test set					Opus-parsebank-test				
Label	Prec	Rec	F	Support	Label	Prec	Rec	F	Support
2	40.2	32.9	36.2	161	neg	95.0	75.0	83.8	6712
3	59.3	52.6	55.8	2434	3	25.2	36.3	29.8	1146
4<	56.0	58.1	57.0	2003	4<	44.7	62.4	52.1	425
4>	58.3	59.8	59.1	2287	4>	46.0	68.0	54.9	560
4	70.5	73.9	72.2	3586	4	56.3	89.7	69.2	793
i	51.8	48.9	50.3	454	i	56.0	71.3	62.7	164
s	49.4	37.7	42.8	438	s	32.0	48.0	38.4	50
W. avg	57.9	58.3	58.0		W. avg	78.1	69.9	72.6	
Acc			58.3		Acc			69.9	

seem to decrease the performance of positive predictions. However, in the opus-parsebank-test, where more than two-thirds of the examples are negatives and therefore larger differences can be expected, the bootstrapped model significantly outperforms the baseline model on the negative class, increasing the negative class F-score from 37.5 to 83.8, which is mostly caused by heavily increasing its recall without compromising the precision too much. This naturally also increases the precision of the positive classes by not as heavily overpredicting the positives; however, the classifier still struggles in distinguishing between different positive labels, as well as precisely setting the border between negatives and contextual paraphrases. When compared with the estimated human performance on the task, the classifier is still almost 12pp behind the accuracy of the human annotators when measured on the Turku Paraphrase Corpus test set. However, in contrast to the humans, the current model does not have access to the document context, which may naturally complicate the labeling decision particularly in the context dependent cases (label 3). In the future, we plan to extend the classification work towards context-aware models.

## 8. Fine-tuned sentence embeddings in paraphrase mining

Paraphrase classification has been shown to work well and is expected to give good results accuracy-wise when judging the paraphrasability of a candidate pair of statements. However, the pair-wise classification approach becomes infeasible especially in large-scale paraphrase retrieval applications, as it requires applying the computationally heavy classifier separately for each possible candidate pair. In large-scale scenarios such as paraphrase mining where the objective is to find good paraphrase candidates from a large collection of sentences, the number of candidate pairs is quadratic. Therefore, computationally a much more feasible approach is to pre-compute sentence embeddings once, and for each candidate pair apply only a computationally light-weight metric (e.g. cosine similarity or Euclidean distance) using these pre-calculated representations. In addition to directly applying a pre-trained language model such as BERT, one can also optimize the representations for paraphrase comparison by fine-tuning these models to create sentence embeddings such that paraphrased statements receive a high similarity score when comparing the calculated embeddings using for example cosine similarity, while semantically unrelated statements receive a low similarity score.

A well-known model of this kind is the Sentence-BERT (SBERT) (Reimers and Gurevych 2019), where the training objective is to improve individual sentence embeddings in order to better support their direct cosine similarity comparison. The SBERT fine-tuning objective applies a siamese network encoding, where the two sentences are first encoded individually producing a fixed size embedding for both, and these embeddings are then fine-tuned through either a classification or cosine similarity objective. The SBERT models are typically trained on semantically related sentences taken from corpora gathered for for example paraphrasing, natural language inference or translation, where the positive pairs are mixed with unrelated sentence pairs in order to provide also negative training examples.

Next, we train a Finnish SBERT model for the paraphrasing task and evaluate it on the task of paraphrase retrieval using the corpus data. In addition to the paraphrase corpus, we evaluate the fine-tuned embedding model also in a large-scale paraphrase mining experiment using a dataset of almost 400M candidate sentences.

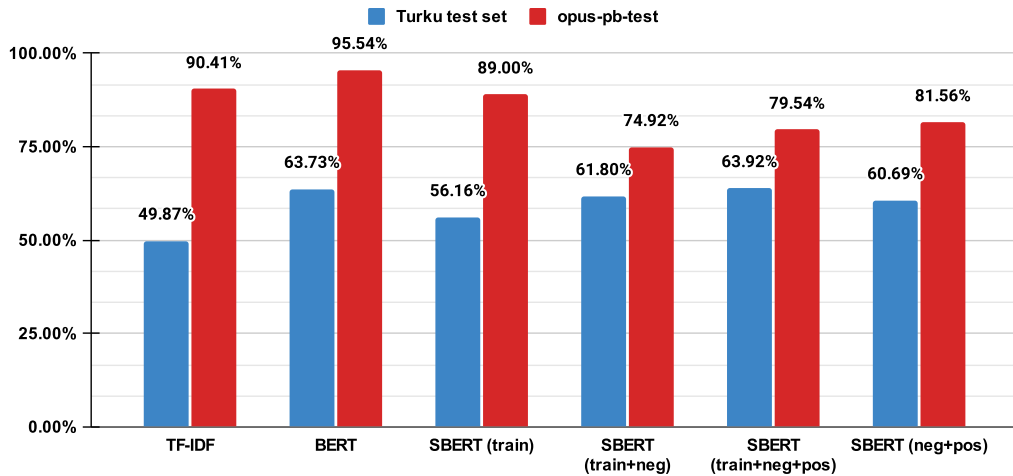
### 8.1. SBERT training and evaluation

In the following, we evaluate the SBERT sentence embedding model on our corpus in the context of paraphrase mining. We train a Finnish SBERT model initialized from the pre-existing Finnish BERT-base model with our paraphrase data. We use batch size of 16 and mean pooling over the final BERT layer, the best-performing pooling method in the original SBERT work (Reimers and Gurevych 2019). Since the goal is to identify paraphrase candidates, we collapse the labels into binary: labels 1 and 2 becomes negative, and labels 3 and above positive. We experiment with different combinations of training datasets: (1) the manually annotated Turku Paraphrase Corpus training set (`train`), consisting of 81.8K positive and 1.4K negative pairs, (2) the manually annotated training set and the full set of automatically gathered negatives (`train+neg`), with 81.8K positives and 5.6M negatives, (3) the manually annotated training set and the full sets of automatically gathered positives and negatives (`train+neg+pos`), totaling 625K positives and 5.6M negatives, and (4) only the automatically gathered positives and negatives (`neg+pos`), with 543K positives and 5.6M negatives. The learning rate is optimized on the development set, using the value  $1e-5$  for all four experiments.

As a non-neural baseline, we use TF-IDF representations of character bi- and tri-grams. As a modern, neural baseline, we use the vanilla Finnish BERT model to directly encode single sentences without any task specific fine-tuning. For hyperparameter optimization, we test CLS vector, mean-pooling, and max-pooling on the development set, and select mean-pooling as the final pooling method.

We evaluate these models on the paraphrase retrieval task, that is given the statement  $s_1$  from a known paraphrase pair  $(s_1, s_2)$ , how well the model is able to identify its corresponding paraphrased version  $s_2$  from a collection of Finnish sentences using cosine similarity. First, we evaluate the retrieval among all paraphrase statements in the corresponding manually annotated test sets. That is, we take both statements from all paraphrase pairs in the corpus test set and deduplicate them. This gives 19,893 unique statements in the Turku Paraphrase Corpus test set and 19,271 in the opus-parsebank-test set. All these candidate text segments are first embedded, and separately for each paraphrase statement in the test set, the candidates are sorted in descending order based on cosine similarity giving the most similar candidates first. A good embedding model is expected to give higher cosine similarity for a paraphrase pair than for a random segment pair, that is rank the known paraphrase pair high in the sorted candidates.

First we measure top-1 retrieval accuracy of all positive examples (labels 3 and above). This is to inspect how likely the model ranks a good paraphrase pair first among candidate sentences if the corresponding paraphrased version is guaranteed to exist in the collection. The results are given in Figure 8. When measured on the Turku Paraphrase Corpus test set (blue color in the figure),



**Figure 8.** The top-1 retrieval accuracy (higher is better) of all positive paraphrases in the Turku Paraphrase Corpus test set and the opus-parsebank-test set. The test sets consists of 19,893 and 19,271 unique retrieval candidates respectively. The exact accuracy numbers are visualized on top of the bars.

the SBERT model `train+neg+pos` gives comparable, if not slightly better, results to the vanilla BERT baseline. The other SBERT models underperform vanilla BERT in terms of top-1 accuracy. Unsurprisingly, all the neural models outperform the TF-IDF method. When considering the opus-parsebank-test set, where the paraphrase candidates were sampled based on a combination of the TF-IDF and FinBERT similarity scores, it is not a surprise to see that these two methods obtain the highest performance. While all examples in the opus-parsebank-test set are selected based on their high BERT similarity score, fully explaining the high top-1 accuracy of the BERT model, the sample was stratified to include examples from all lexical similarity areas. However, after manual annotation most of the positive examples are actually located in the high similarity area (the average lexical similarity of positive examples being 0.73 compared with 0.5 on the full development sample), therefore to some extent skewing the evaluation also in terms of TF-IDF similarity.

However, measuring the top-1 accuracy of the positive paraphrases does not take into consideration how these models perform on the negative pairs, where the model should not give a high similarity for nonparaphrase pairs even if their lexical similarity is high. In the light of this, we next measure the average ranking positions of the paraphrase candidates separately for each label in order to see whether fine-tuning the model successfully decreases the similarity of negative paraphrase pairs while increasing or maintaining the similarity of the positive pairs, as it is expected that a good model should give lower similarity and therefore also worse ranking positions for unrelated candidates than for related candidates, while similarity of related candidates in turn should be lower than similarity of real paraphrases and so on. To measure this effect, in Figures 9 and 10, we report average ranking positions in percentage for each label separately, where the actual on average ranking positions are normalized to percentages, so as there were 100 candidates. This means that a perfect ranking, where the correct candidate is always ranked top-1, would give 0%, whereas the results of 5% means that the correct candidate is on average ranked 6th out of 100 candidates.

Based on the results in Figure 9, our ranking assumption seems to hold in the sense that the more universal the paraphrase pair is the better the average ranking position seems to be in general. However, with the exception of the vanilla BERT and SBERT trained on the Turku Paraphrase



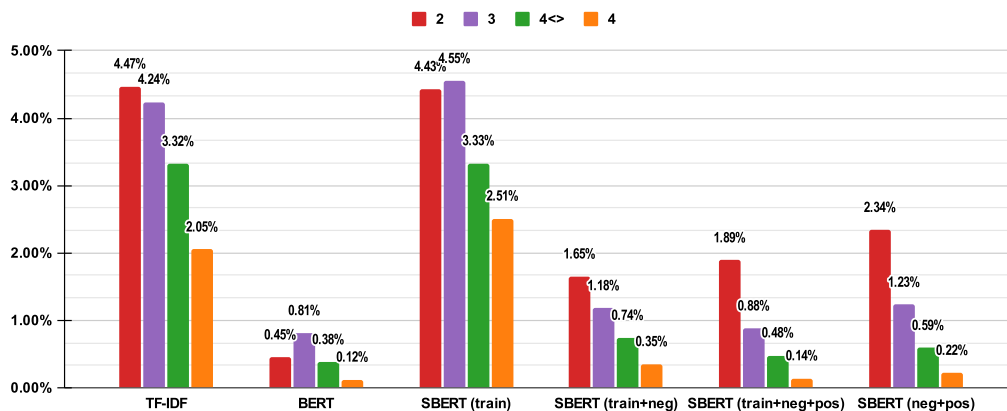


Figure 9. The average ranking positions normalized to percentages (lower is better) for the Turku Paraphrase Corpus test set by various models. The ranking is measured separately for each paraphrase label (2, 3, 4<->, and 4), however disregarding the flags i and s. The exact numbers are visualized on top of the bars (percentage calculated out of 19,893 candidate sentences).

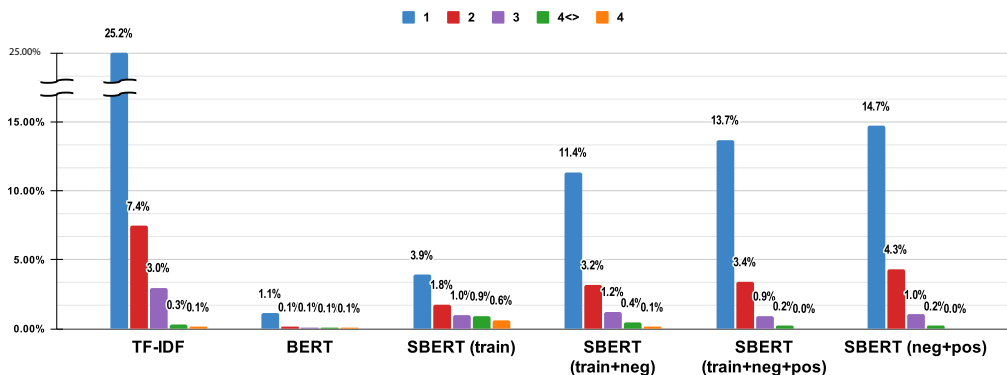


Figure 10. The retrieval of the opus-parsebank test set paraphrase candidates by various models. The numbers on top of the bars indicate the average ranking in percentage (out of 19,271 candidate sentences) for each class of paraphrase candidates. The ranking is measured separately for each paraphrase label (1, 2, 3, 4<->, and 4), however disregarding the flags i and s.

Corpus only (train, where fine-tuning data does not include practically at all negatives) models do not distinguish between the negative label 2 and positive label 3, therefore giving high similarity scores also for negative paraphrase pairs. However, when increasing the amount of negative examples seen during the training, the fine-tuned SBERT models start to give clearly worse ranking positions for label 2 pairs compared with label 3 pairs as the model learns to judge these as negative examples, which appears to be the main advantage of SBERT models over the vanilla BERT. When comparing the different SBERT models, the observations remain largely the same as in the top-1 accuracy analysis. That is, the SBERT model trained with all available training data yielding the best results among the fine-tuned models. For the evaluation on the opus-parsebank-test set (Figure 10), the average of BERT embeddings clearly achieves the best ranking positions, which is not at all surprising as the test set was selected based on the similarity of BERT embeddings. Again, the notable fact is that while the original BERT naturally assigns good ranking positions for the negative examples in this dataset as well (label 1 and 2), the model fine-tuning clearly helps to distinguish between positive and negative examples, pushing the negative examples further while only negligibly affecting the ranking for the positive examples.

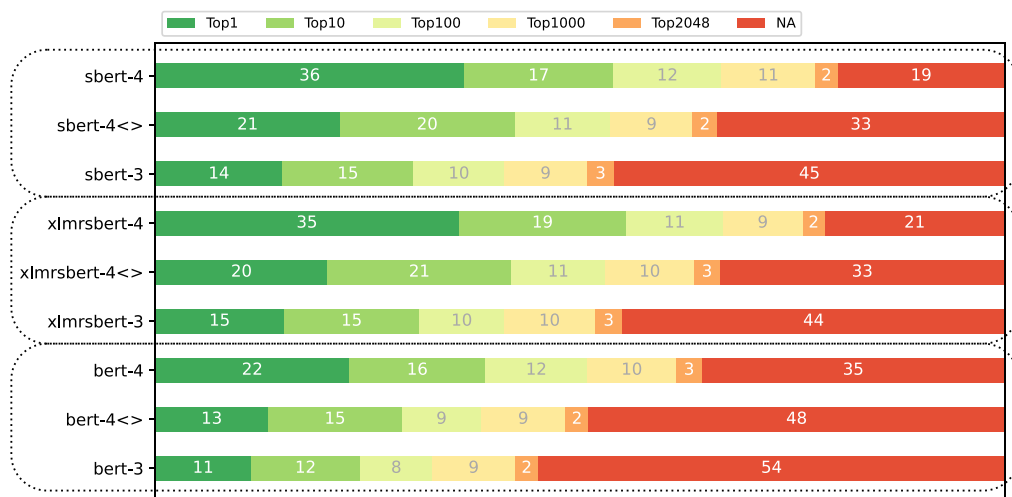
## 8.2. Large-scale paraphrase mining

A larger collection of Finnish candidate sentences presumably makes the paraphrase mining task more difficult as the number of difficult distractors also increases. For instance, considering top-1 accuracy, it takes only one incorrect distractor sentence to fool the model. Thus, we simulate a realistic paraphrase mining setting by mining the correct target sentence among the combined set of 399M unique sentences from the combination of the Finnish Internet Parsebank, OPUS, and our paraphrase corpus. First, we calculate and index the SBERT embedding for each sentence in this large combined dataset. Then, for each test set paraphrase pair ( $s_1, s_2$ ), we query the index with the embedding of  $s_1$  and measure at which rank out of the nearly 400M candidates the embedding of  $s_2$  is found in terms of Euclidean distance. For comparison, we also carry out the same experiment with the vanilla FinBERT model embeddings, so as to establish whether the fine-tuning of the SBERT model translates into better performance on the sentence similarity task, as well as with the multilingual SBERT model `paraphrase-xlm-r-multilingual-v1` released by Reimers and Gurevych (2019) fine-tuned to create comparable embeddings for over 50 languages. The multilingual SBERT model is based on the monolingual English SBERT trained on a massive collection of semantically similar English sentence pairs, and the multilingual XLM-RoBERTa-base language model (Conneau *et al.* 2020), where the multilingual language model was fine-tuned to mimic the embeddings of the English SBERT using multilingual knowledge distillation (teacher–student framework) on parallel data for over 50 languages.

The results are summarized in Figure 11, where we report the top- $N$  accuracy (where  $N=1, 10, 100, 1000$ , and 2048, which is the upper technical limit in the experiment) for label 3, label 4> or 4<, and label 4 separately. Most importantly, for the Finnish SBERT model (named `sbert` in the figure), we can see that 53% of label 4 paraphrases, 41% of label 4> or 4< paraphrases, and 29% of label 3 paraphrases are ranked among the top 10 most similar sentences from the group of nearly 400M candidates. This demonstrates that the SBERT model is highly efficient at finding paraphrase pairs also in cases where the number of candidates is in the hundreds of millions. This opens the possibility for further paraphrase mining from even very large text collections. While it is obviously infeasible to apply an expensive pairwise classification model to all sentence pairs (in our case that would be on the order of 400M squared), one can use SBERT as an initial filter and then apply the pairwise classification model to the comparatively small number of top candidates (in our case 400M times 10 pairs if using the cut-off of top-10 candidates). Finally, as seen in Figure 11, the vanilla FinBERT (`bert` in the figure) not fine-tuned for the semantic similarity task produces notably worse results compared with the SBERT models, while both Finnish and multilingual (`xmlrsbert` in the figure) SBERT produce comparable results. This seems to indicate that the advantage of model fine-tuning starts to pay off when the number of candidates for the retrieval is substantially increased. With this massive candidate set, the SBERT models are likely better at filtering out topically and lexically difficult distractors, which did not show up when using a smaller candidate set. The implementation of this experiment was carried out using the FAISS library (Johnson *et al.* 2021) for efficient GPU-accelerated k-nearest-neighbor vector similarity search in large vector collections.

## 9. Conclusions

In this paper, we presented the Turku Paraphrase Corpus, the first large-scale manually annotated corpus of Finnish paraphrases. The corpus contains 104,645 paraphrase pairs, targeted to create a challenging paraphrasing dataset suitable to test the capabilities of natural language understanding models. Each pair is manually labeled using a detailed annotation scheme. In addition to separating positive and negative paraphrase pairs, the annotation also distinguishes between paraphrases in all imaginable contexts and paraphrases in the given context but not necessarily elsewhere.



**Figure 11.** The retrieval of test set paraphrase pairs by the fine-tuned Finnish SBERT, the multilingual SBERT, and the vanilla FinBERT, out of 400M candidate sentences. The white numbers indicate percentage of pairs in the given category, and the retrieval is measured for the three main classes of paraphrase: 4, 4< or 4>, and 3 (disregarding flags s and i); and for several top  $k$  cut-offs. NA means that the correct sentence did not rank in the top 2048 list, which was the upper technical limit in the experiment.

The paraphrase pairs in the corpus are collected using a novel method for manual paraphrase candidate extraction, assuring both quality and variability of the extracted paraphrases, as well as efficiency in terms of person-months used for annotation. The paraphrases are manually selected from two related source documents, where a high tendency of naturally occurring paraphrases is expected. Compared with other paraphrase resources, the manual extraction is shown to produce notably longer and less lexically overlapping pairs than what automated candidate selection permits, creating a challenging dataset to be used for instance in evaluation of different language understanding models. In addition to quality, the advantage of manual candidate extraction is the possibility to collect and evaluate the paraphrase candidates in their original document context, setting many new possibilities for contextual paraphrase recognition. To our knowledge, this work is the first large-scale paraphrase corpus providing original document context information for the paraphrase pairs.

While 98% of the paraphrases in the corpus are manually classified to be at least paraphrases in their given context if not in all contexts (positive examples), in order to better facilitate also binary classification experiments (paraphrase or not-a-paraphrase), a method for semi-automatically extracting negative paraphrase candidates is presented, and a supplementary set of over 5 million negative paraphrase candidates is provided together with the actual corpus.

The initial modeling results confirmed the challenging nature of the dataset, giving weighted mean F-score of 58% for a pairwise classifier over the detailed annotation labels, the classifier accuracy substantially lacking behind the estimated human performance on the task. However, when applying semantic similarity models fine-tuned on the data for large-scale paraphrase mining from a collection of almost 400M candidates, the results were highly encouraging, the paraphrase retrieval model being able to rank the correct paraphrase pair among the top-10 for 29–53% of the evaluation examples depending on the paraphrase type.

While our initial paraphrase retrieval experiments show promising results, the classification experiments using the detailed labeling scheme are still far from human performance, indicating that the corpus can serve as a challenging evaluation task for different language understanding models. Such datasets have recently shown their importance when yet more powerful language

understanding models are approaching human-level performance on several popular evaluation sets, and more challenging tasks are introduced (Wang *et al.* 2019). However, despite our initial modeling experiments, there are still many new aspects to study with the dataset, such as how to utilize the contextual information available for the paraphrase pairs, and in the future work, we plan to further study the contextuality aspect of this data.

The corpus is available at [github.com/TurkuNLP/Turku-paraphrase-corpus](https://github.com/TurkuNLP/Turku-paraphrase-corpus) as well as through the popular HuggingFace datasets under the CC-BY-SA license.

**Acknowledgements.** We warmly thank Leena Salmi, Eriikka Paavilainen-Mäntymäki, Riikka Harikkala-Laihininen and Veronika Laippala for their support in student data collection, as well as all anonymous students for agreeing to share their data. We gratefully acknowledge the support of European Language Grid which funded the annotation work. Computational resources were provided by CSC—the Finnish IT Center for Science and the research was supported by the Academy of Finland and the Digicampus project. We also thank Sampo Pyysalo for fruitful discussions and feedback throughout the project and Jörg Tiedemann for his generous assistance with the OpenSubtitles data.

**Conflicts of interest.** The authors declare none.

## References

- Altheneyan A.S. and Menai M.E.B.** (2019). Evaluation of state-of-the-art paraphrase identification and its application to automatic plagiarism detection. *International Journal of Pattern Recognition and Artificial Intelligence* **34**(4). <https://doi.org/10.1142/S0218001420530043>
- Arwinder Singh G.S.J.** (2020). Construction of paraphrasing dataset for Punjabi: A deep learning approach. *International Journal of Advanced Science and Technology* **29**(06), 9433–9442.
- Bhagat R. and Hovy E.** (2013). Squibs: What is a paraphrase? *Computational Linguistics* **39**(3), 463–472.
- Chang L.-H., Pyysalo S., Kanerva J. and Ginter F.** (2021). Quantitative evaluation of alternative translations in a corpus of highly dissimilar Finnish paraphrases. In *Proceedings for the First Workshop on Modelling Translation: Translatology in the Digital Age* online. Association for Computational Linguistics, pp. 100–107.
- Conneau A., Khandelwal K., Goyal N., Chaudhary V., Wenzek G., Guzmán F., Grave E., Ott M., Zettlemoyer L. and Stoyanov V.** (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics, pp. 8440–8451.
- Creutz M.** (2018). Open Subtitles paraphrase corpus for six languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA), pp. 1364–1369.
- Davani A.M., Daz M. and Prabhakaran V.** (2022). Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics* **10**, 92–110.
- Devlin J., Chang M.-W., Lee K. and Toutanova K.** (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics, pp. 4171–4186.
- Dolan W.B. and Brockett C.** (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP 2005)*, Jeju Island, Korea. Asian Federation of Natural Language Processing, pp. 9–16.
- Dong Q., Wan X. and Cao Y.** (2021). ParaSCI: A large scientific paraphrase dataset for longer paraphrase generation. arXiv preprint arXiv:2101.08382.
- Eyecioglu A. and Keller B.** (2018). Constructing a Turkish corpus for paraphrase identification and semantic similarity. In **Gelbukh A.** (ed), *Computational Linguistics and Intelligent Text Processing*, Cham. Springer International Publishing, pp. 588–599.
- Federmann C., Elachqar O. and Quirk C.** (2019). Multilingual whispers: Generating paraphrases with translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, Hong Kong, China. Association for Computational Linguistics, pp. 17–26.
- Ganitkevitch J. and Callison-Burch C.** (2014). The multilingual paraphrase database. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA), pp. 4276–4283.
- Ganitkevitch J., Van Durme B. and Callison-Burch C.** (2013). PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia. Association for Computational Linguistics, pp. 758–764.

- Gudkov V., Mitrofanova O. and Filippikh E.** (2020). Automatically ranked Russian paraphrase corpus for text generation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, Online. Association for Computational Linguistics, pp. 54–59.
- Guo M., Shen Q., Yang Y., Ge H., Cer D., Hernandez Abrego G., Stevens K., Constant N., Sung Y.-H., Strope B. and Kurzweil R.** (2018). Effective parallel corpus mining using bilingual sentence embeddings. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, Brussels, Belgium. Association for Computational Linguistics, pp. 165–176.
- He Y., Wang Z., Zhang Y., Huang R. and Caverlee J.** (2020). PARADE: A new dataset for paraphrase identification requiring computer science domain knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics, pp. 7572–7582.
- Johnson J., Douze M. and Jégou H.** (2021). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 535–547.
- Kanerva J., Ginter F., Chang L.-H., Rastas I., Skantsi V., Kilpeläinen J., Kupari H.-M., Piirto A., Saarni J., Sevón M. and Tarkka O.** (2021a). Annotation guidelines for the Turku Paraphrase Corpus. Technical report, University of Turku, arXiv:2108.07499.
- Kanerva J., Ginter F., Chang L.-H., Rastas I., Skantsi V., Kilpeläinen J., Kupari H.-M., Saarni J., Sevón M. and Tarkka O.** (2021b). Finnish paraphrase corpus. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden, pp. 288–298.
- Kanerva J., Ginter F., Miekka N., Leino A. and Salakoski T.** (2018). Turku neural parser pipeline: An end-to-end system for the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium. Association for Computational Linguistics, pp. 133–142.
- Lan W., Qiu S., He H. and Xu W.** (2017). A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark. Association for Computational Linguistics, pp. 1224–1234.
- Luotolahti J., Kanerva J., Laippala V., Pyysalo S. and Ginter F.** (2015). Towards universal web parsebanks. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, Uppsala, Sweden. Uppsala University, Uppsala, Sweden, pp. 211–220.
- Mehdizadeh Seraj R., Siahbani M. and Sarkar A.** (2015). Improving statistical machine translation with a multilingual paraphrase database. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal. Association for Computational Linguistics, pp. 1379–1390.
- Pavlick E. and Kwiatkowski T.** (2019). Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics* 7, 677–694.
- Pivovarova L., Pronoza E., Yagunova E. and Pronoza A.** (2018). Paraphraser: Russian paraphrase corpus and shared task. In Filchenkov A., Pivovarova L. and Žižka J. (eds), *Artificial Intelligence and Natural Language*, Cham. Springer International Publishing, pp. 211–225.
- Reimers N. and Gurevych I.** (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics, pp. 3982–3992.
- Scherrer Y.** (2020). TaPaCo: A corpus of sentential paraphrases for 73 languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association, pp. 6868–6873.
- Schwenk H. and Douze M.** (2017). Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, Vancouver, Canada. Association for Computational Linguistics, pp. 157–167.
- Shimohata M., Sumita E. and Matsumoto Y.** (2004). Building a paraphrase corpus for speech translation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA), pp. 1407–1410.
- Soni S. and Roberts K.** (2019). A paraphrase generation system for EHR question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, Florence, Italy. Association for Computational Linguistics, pp. 20–29.
- Tiedemann J.** (2012). Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA), pp. 2214–2218.
- Tiedemann J.** (2020). The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics, pp. 1174–1182.
- Virtanen A., Kanerva J., Ilo R., Luoma J., Luotolahti J., Salakoski T., Ginter F. and Pyysalo S.** (2019). Multilingual is not enough: BERT for Finnish. arXiv preprint arXiv:1912.07076.

- Wang A., Pruksachatkun Y., Nangia N., Singh A., Michael J., Hill F., Levy O. and Bowman S.** (2019). SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E. and Garnett R. (eds), *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc.
- Wieting J. and Gimpel K.** (2018). ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics, pp. 451–462.

---

**Cite this article:** Kanerva J, Ginter F, Chang L-H, Rastas I, Skantsi V, Kilpeläinen J, Kupari H-M, Piirto A, Saarni J, Sevón M and Tarkka O. Towards diverse and contextually anchored paraphrase modeling: A dataset and baselines for Finnish. *Natural Language Engineering* <https://doi.org/10.1017/S1351324923000086>





**TURUN  
YLIOPISTO**  
UNIVERSITY  
OF TURKU

ISBN 978-951-29-9622-3 (PRINT)  
ISBN 978-951-29-9623-0 (PDF)  
ISSN 2736-9390 (PRINT)  
ISSN 2736-9684 (ONLINE)