Jari Björne

Biomedical Event Extraction with Machine Learning

Turku Centre *for* Computer Science

TUCS Dissertations

No 178, July 2014

Biomedical Event Extraction with Machine Learning

Jari Björne

To be presented, with the permission of the Faculty of Mathematics and Natural Sciences of the University of Turku, for public criticism in Auditorium Beta of the ICT-Building on August 7, 2014, at 12 noon.

University of Turku
Department of Information Technology
FI-20014 Turun yliopisto
Finland

2014

Supervisors

Professor Tapio Salakoski
Department of Information Technology
University of Turku
Finland

Doctor Filip Ginter
Department of Information Technology
University of Turku
Finland

Reviewers

Docent Jussi Karlgren
KTH Royal Institute of Technology
Stockholm
Sweden

Doctor Fabio Rinaldi
Institute of Computational Linguistics
University of Zurich
Switzerland

Opponent

Doctor Pierre Zweigenbaum
LIMSI-CNRS
University of Paris-Sud
France

ISBN 978-952-12-3078-3
ISSN 1239-1883

Abstract

Biomedical natural language processing (BioNLP) is a subfield of natural language processing, an area of computational linguistics concerned with developing programs that work with natural language: written texts and speech. Biomedical relation extraction concerns the detection of semantic relations such as protein–protein interactions (PPI) from scientific texts. The aim is to enhance information retrieval by detecting relations between concepts, not just individual concepts as with a keyword search.

In recent years, events have been proposed as a more detailed alternative for simple pairwise PPI relations. Events provide a systematic, structural representation for annotating the content of natural language texts. Events are characterized by annotated trigger words, directed and typed arguments and the ability to nest other events. For example, the sentence "Protein A causes protein B to bind protein C" can be annotated with the nested event structure *CAUSE(A, BIND(B, C))*. Converted to such formal representations, the information of natural language texts can be used by computational applications. Biomedical event annotations were introduced by the BioInfer and GENIA corpora, and event extraction was popularized by the BioNLP'09 Shared Task on Event Extraction.

In this thesis we present a method for automated event extraction, implemented as the Turku Event Extraction System (TEES). A unified graph format is defined for representing event annotations and the problem of extracting complex event structures is decomposed into a number of independent classification tasks. These classification tasks are solved using SVM and RLS classifiers, utilizing rich feature representations built from full dependency parsing. Building on earlier work on pairwise relation extraction and using a generalized graph representation, the resulting TEES system is capable of detecting binary relations as well as complex event structures.

We show that this event extraction system has good performance, reaching the first place in the BioNLP'09 Shared Task on Event Extraction. Subsequently, TEES has achieved several first ranks in the BioNLP'11 and BioNLP'13 Shared Tasks, as well as shown competitive performance in the binary relation Drug-Drug Interaction Extraction 2011 and 2013 shared tasks.

The Turku Event Extraction System is published as a freely available open-source project, documenting the research in detail as well as making the method available for practical applications. In particular, in this thesis we describe the application of the event extraction method to PubMed-scale text mining, showing how the developed approach not only shows good performance, but is generalizable and applicable to large-scale real-world text mining projects.

Finally, we discuss related literature, summarize the contributions of the work and present some thoughts on future directions for biomedical event extraction. This thesis includes and builds on six original research publications. The first of these introduces the analysis of dependency parses that leads to development of TEES. The entries in the three BioNLP Shared Tasks, as well as in the DDIExtraction 2011 task are covered in four publications, and the sixth one demonstrates the application of the system to PubMed-scale text mining.

Acknowledgements

First, I would like to thank my supervisors Tapio Salakoski and Filip Ginter. They have guided my work throughout this thesis and before, starting from 2004 when I had my first summer job in professor Salakoski's group. They have provided direction, support and advice, but have also allowed me the independence so important for scientific research. While not a supervisor on this thesis, Sampo Pyysalo has also had a big impact on my career. From the early work on the BioInfer corpus to the massive PubMed-scale text mining projects, Filip and Sampo have taught me most everything I know of natural language processing and computational linguistics. I am also grateful for their rigorous approach to science, and their curiosity and ambition in finding the next big challenges to tackle. Finalizing a PhD thesis can be a complex task with many unforeseen issues, so I would also like to thank postgraduate studies coordinator Maritta Löytömäki for her quick and reliable support whenever seemingly insurmountable obstacles appeared to delay the finishing of this work.

Antti Airola and Tapio Pahikkala have been instrumental in teaching me how to use machine learning, and in providing the experience and theoretical grounding for its efficient application in my work. My first experience with machine learning was when participating in Antti's graph kernel project in 2008, which was also the first time I was involved in writing a larger software project. I thank Antti for the great collaboration and for what was also a central building block for the ideas that would later become the Turku Event Extraction System. I would like to thank Juho Heimonen for collaborations on event extraction and our many discussions on potential approaches to biomedical text mining and the annotations that make such things possible.

Regarding the reviews of this thesis, I would like to thank Doctors Jussi Karlgren and Fabio Rinaldi for their insightful comments. I am in particular grateful to Dr. Karlgren's view as an "outsider" to biomedical event extraction, highlighting several important issues and providing an understanding of how this work might be viewed in the larger context of natural language processing. I am most thankful to Doctor Pierre Zweigenbaum for agreeing to act as my opponent.

It is also important to note the contribution of the TUCS graduate school in providing me with the opportunity to focus full-time on this PhD research. The importance of continued, reliable research funding for the duration of one's graduate studies cannot be overstated. TUCS has also contributed numerous travel grants allowing me to attend several highly ranked scientific conferences, to present my work in an international setting and to become connected with the global scientific community. I am very grateful to the TUCS board for their continued support of my research, and Tomi Mäntylä, Irmeli Laine and all the staff of TUCS for their guidance and assistance throughout my graduate studies. I would also like to thank the Turku University Foundation for providing travel grants.

In 2010 I had the unique opportunity to visit the University of Tokyo for a three-month research exchange, and I wish to warmly thank Professor Jun'ichi Tsujii for this wonderful experience. The Tsujii Laboratory is the source of many of the largest and most impactful projects in the BioNLP field, including the GENIA corpus, but it was also a great pleasure to find it such a vibrant, active and accepting environment, truly a model of what a great research lab can be. In addition to professor Tsujii I would like to thank Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim, Rune Sætre, Yoshinobu Kano, Han-Cheol Cho, Pontus Stenetorp and Goran Topić for their guidance and advice on my questions regarding biomedical text mining and practical programming tasks, and all the Tsujii-lab members with whom I shared many interesting scientific discussions and pleasurable social occasions. I would also like to thank Noriko Katsu for her thorough and timely work in organizing the travel and other arrangements of my exchange. My visit would not have been possible without the grants of the Academy of Finland and the The Scandinavia-Japan Sasakawa Foundation, whose boards I wish to thank for their support.

My work on this thesis has also been supported by the scholarships of the Nokia Foundation. In addition to their generous support, I would like to thank the Nokia Foundation for their yearly awards ceremony, which brings a more personal aspect to their support and provides a nice cross-disciplinary meeting for students of many different aspects of information technology. The CSC – IT Center for Science Ltd has provided the cluster computing environments used in much of the research of this thesis. Without these computational resources much of the work, especially the PubMed-scale event extraction, would simply not have been possible. Having the supercomputing resources of CSC available for all Finnish universities is a wonderful form of support for many and diverse forms of computer sciences research.

The Department of Information Technology at the University of Turku has provided a reliable environment in which to pursue my PhD studies.

Starting to work there full-time in 2007, by virtue of limited space and random chance, I ended up sitting in a room with people working on very different projects from my own. I would like to thank Erkki Kaila, Peter Larsson and Teemu Rajala for their daily companionship and support, and for making this work that has often been a solitary project into a much more human and enjoyable experience. Being exposed to research questions and projects outside one's immediate field has also had a very positive impact on my work.

The BioNLP research at the IT department continues to advance and recruit new talent, and it has been great to see our newer members Suwisa Kaewphan, Kai Hakala and Farrokh Mehryary develop into capable researchers and start to pursue their own aspects of BioNLP research. I would like to thank them for their collaborations and feedback on many aspects of the research presented in this book, too. I would also like to thank our international visitors, Sofie van Landeghem and Hans Moen for bringing many new ideas and viewpoints to this group. In particular, I thank Sofie for our multiple succesful collaborations on joint publications.

Much of the work in this thesis has been tested in *shared tasks*, international competitive evaluations where any researcher can participate and propose a solution to a given scientific problem. The opportunity to test a diverse set of potential approaches through an impartial, objective metric, free from the human biases and preconceptions that can exist in any scientific endeavour, has been a valuable contribution to the field of biomedical text mining. Organizing such tasks is a huge effort, and I wish to express my gratitude for everyone who has made them possible. The BioNLP Shared Task on Event Extraction was originally, in 2009, organized by the Tsujii Laboratory of the University of Tokyo, but has since then expanded into a broad international collaboration involving many teams and diverse aspects of biomedical text mining. I would like to thank everyone involved in organizing the BioNLP Shared Tasks for all their work and for continuously pushing event extraction towards new and interesting domains. The DDIExtraction shared task considers the detection of drug–drug interactions, and has provided an important opportunity to connect my work on TEES and event extraction to the large, related domain of relation extraction, while also evolving the field of binary relation extraction. I would like to thank its organizers Isabel Segura-Bedmar, Paloma Martínez, Daniel Sánchez-Cisneros and María Herrero-Zazo for what has always been a well-focused and efficient shared task.

Having open-sourced the Turku Event Extraction System, it has been a truly wonderful experience to see the program become useful for many different research projects and applications across the world. For the many users of TEES, I apologize for my occasionally late email replies and for the continuous work-in-progress nature of the code, and wish to express how

deeply grateful I am for all your contacts and kind words regarding this work. The very positive reception TEES has received has further convinced me how very important it is to open-source the codes behind our research.

Working on a PhD can often be an all-consuming experience, and can occasionally threaten to take over one's entire life with the constant pressures and deadlines. I would like to thank my friends and family for their support and the all-important opportunities to step outside the academic bubble. My parents Aira and Lars have been an unwavering support on this project, and without their love and encouragement throughout the years it is unlikely this book would have been written.

Turku, July 2014
Jari Björne

List of original publications

I Björne, J., Pyysalo, S., Ginter, F., and Salakoski, T. (2008). How Complex are Complex Protein-protein Interactions? In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM08)*, pages 125–128. Turku Centre for Computer Science (TUCS)

II Björne, J., Heimonen, J., Ginter, F., Airola, A., Pahikkala, T., and Salakoski, T. (2011b). Extracting contextualized complex biological events with rich graph-based feature sets. *Computational Intelligence*, 27(4):541–557

III Björne, J., Ginter, F., Pyysalo, S., Tsujii, J., and Salakoski, T. (2010a). Complex event extraction at PubMed scale. *Bioinformatics*, 26(12):i382–i390

IV Björne, J., Ginter, F., and Salakoski, T. (2012a). University of Turku in the BioNLP'11 Shared Task. *BMC bioinformatics*, 13(Suppl 11):S4

V Björne, J., Airola, A., Pahikkala, T., and Salakoski, T. (2011a). Drug-Drug Interaction Extraction from Biomedical Texts with SVM and RLS Classifiers. In *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction 2011*, pages 35–42. CEUR Workshop Proceedings

VI Björne, J. and Salakoski, T. (2013). TEES 2.1: Automated annotation scheme learning in the BioNLP 2013 Shared Task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 16–25, Sofia, Bulgaria. Association for Computational Linguistics

List of co-authored publications not included in the thesis

- Pyysalo, S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., and Salakoski, T. (2007). BioInfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, 8(1):50

- Ginter, F., Pyysalo, S., Björne, J., Heimonen, J., and Salakoski, T. (2007). BioInfer relationship annotation manual. Technical report, Technical Report 806, Turku Centre for Computer Science

- Pyysalo, S., Airola, A., Heimonen, J., Björne, J., Ginter, F., and Salakoski, T. (2008). Comparative analysis of five protein-protein interaction corpora. *BMC bioinformatics*, 9(Suppl 3):S6

- Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., and Salakoski, T. (2008a). A graph kernel for protein-protein interaction extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 1–9. Association for Computational Linguistics

- Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., and Salakoski, T. (2008b). All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics*, 9(Suppl 11):S2

- Björne, J., Ginter, F., Heimonen, J., Pyysalo, S., and Salakoski, T. (2009a). Learning to extract biological event and relation graphs. *Proceedings of the 17th Nordic Conference on Computational Linguistics (NODALIDA09)*

- Björne, J., Heimonen, J., Ginter, F., Airola, A., Pahikkala, T., and Salakoski, T. (2009b). Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the Workshop on Current*

Trends in Biomedical Natural Language Processing: Shared Task, pages 10–18. Association for Computational Linguistics

- Björne, J., Ginter, F., Pyysalo, S., Tsujii, J., and Salakoski, T. (2010b). Scaling up biomedical event extraction to the entire PubMed. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 28–36. Association for Computational Linguistics

- Heimonen, J., Björne, J., and Salakoski, T. (2010). Reconstruction of semantic relationships from their projections in biomolecular domain. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 108–116. Association for Computational Linguistics

- Ginter, F., Björne, J., and Pyysalo, S. (2010). Event extraction on PubMed scale. *BMC Bioinformatics*, 11(Suppl 5):O2

- Björne, J. and Salakoski, T. (2011b). Generalizing biomedical event extraction. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 183–191, Portland, Oregon, USA. Association for Computational Linguistics

- Björne, J. and Salakoski, T. (2011a). A Machine Learning Model and Evaluation of Text Mining for Protein Function Prediction. In *Automated Function Prediction Featuring a Critical Assessment of Function Annotations (AFP/CAFA) 2011*, pages 7–8. Automated Function Prediction – an ISMB Special Interest Group

- Kano, Y., Björne, J., Ginter, F., Salakoski, T., Buyko, E., Hahn, U., Cohen, K. B., Verspoor, K., Roeder, C., Hunter, L. E., et al. (2011). U-Compare bio-event meta-service: compatible BioNLP event extraction services. *BMC bioinformatics*, 12(1):481

- Björne, J., Van Landeghem, S., Pyysalo, S., Ohta, T., Ginter, F., Van de Peer, Y., Ananiadou, S., and Salakoski, T. (2012b). PubMed-scale event extraction for post-translational modifications, epigenetics and protein structural relations. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, pages 82–90. Association for Computational Linguistics

- Van Landeghem, S., Björne, J., Abeel, T., De Baets, B., Salakoski, T., and Van de Peer, Y. (2012a). Semantically linking molecular entities in literature through entity relationships. *BMC bioinformatics*, 13(Suppl 11):S6

- Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T. W., Sokolov, A., Graim, K., Funk, C., Verspoor, K., Ben-Hur, A.,

Pandey, G., Yunes, J. M., Talwalkar, A. S., Repo, S., Souza, M. L., Piovesan, D., Casadio, R., Wang, Z., Cheng, J., Fang, H., Gough, J., Koskinen, P., Törönen, P., Nokso-Koivisto, J., Holm, L., Cozzetto, D., Buchan, D. W. A., Bryson, K., Jones, D. T., Limave, B., Inamdar, H., Datta, A., Manjari, S. K., Joshi, R., Chitale, M., Kihara, D., Lisewski, A. M., Erdin, S., Venner, E., Lichtarge, O., Rentzsch, R., Yang, H., Romero, A. E., Bhat, P., Paccanaro, A., Hamp, T., Kaner, R., Seemayer, S., Vicedo, E., Schaefer, C., Achten, D., Auer, F., Boehm, A., Braun, T., Hecht, M., Heron, M., Hönigschmid, P., Hopf, T. A., Kaufmann, S., Kiening, M., Krompass, D., Landerer, C., Mahlich, Y., Roos, M., Björne, J., Salakoski, T., Wong, A., Shatkay, H., Gatzmann, F., Sommer, I., Wass, M. N., Sternberg, M. J. E., Škunca, N., Supek, F., Bošnjak, M., Panov, P., Džeroski, S., Šmuc, T., Kourmpetis, Y. A. I., van Dijk, A. D. J., ter Braak, C. J. F., Zhou, Y., Gong, Q., Dong, X., Tian, W., Falda, M., Fontana, P., Lavezzo, E., Camillo, B. D., Toppo, S., Lan, L., Djuric, N., Guo, Y., Vucetic, S. V., Bairoch, A., Linial, M., Babbitt, P. C., Brenner, S. E., Orengo, C., Rost, B., Mooney, S. D., and Friedberg, I. (2013). A Large-Scale Evaluation of Computational Protein Function Prediction. *Nature methods*, 10:221–227

- Van Landeghem, S., Björne, J., Wei, C.-H., Hakala, K., Pyysalo, S., Ananiadou, S., Kao, H.-Y., Lu, Z., Salakoski, T., Van de Peer, Y., et al. (2013a). Large-scale event extraction from literature with multi-level gene normalization. *PloS one*, 8(4):e55814

- Björne, J., Kaewphan, S., and Salakoski, T. (2013). UTurku: Drug Named Entity Recognition and Drug-Drug Interaction Extraction Using SVM Classification and Domain Knowledge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 651–659, Atlanta, Georgia, USA. Association for Computational Linguistics

Contents

Chapter 1

Introduction

Biomedical natural language processing (BioNLP) refers to the automated analysis and extraction of information from texts related to biology and medicine. Natural language processing, NLP, is a field of computational linguistics concerned with developing methods for enabling computers to work with natural language, i.e. written texts or spoken language, the natural forms of human communication. In addition to extending our understanding of how language works, NLP aims to ease and improve human–computer interaction by allowing humans to use more natural forms of communication when interacting with computers.

BioNLP has developed as a subfield of NLP, taking NLP techniques and applying them to biomedical questions. It has emerged as a response to the exponential growth of information in biology, and the need of scientists and medical professionals to extract very specialized information from this mass of knowledge. For example, PubMed[1], the central repository of biomedical research articles, contains over 20 million citations and is growing at an exponential rate. While traditional keyword searches can retrieve a set of articles of interest, often much additional work is required to distill this mass of text into actually relevant knowledge, especially when searches include common concepts. The BioNLP field aims to provide more advanced tools for information retrieval, looking into the semantic relations existing between concepts in biomedical natural language texts.

1.1 Biomedical Event Extraction

Much of modern biology revolves around signaling networks, the complicated relationships formed by the interactions of the molecules that make up cells and larger organisms. The DNA of a genome defines the genes that encode the proteins which regulate and process the complex metabolism of

[1] `https://www.ncbi.nlm.nih.gov/pubmed/`

an organism: the diverse functions ranging from control of cell structure and motility to generation of energy from nutrients and defence against pathogens via the immune system. The scientific publications discussing molecular biology often present complex statements of *semantic relations* (using verbs such as "regulates" to describe metabolic interactions) between *named entities* (nouns referring to genes, proteins and other biological components). Extracting these statements from text to a set of formally defined relations is a case of biomedical natural language processing.

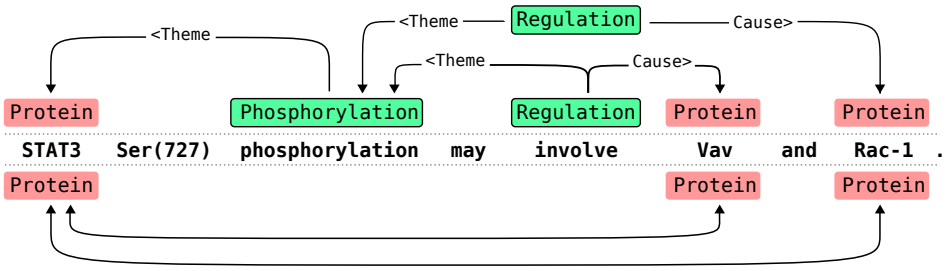
Traditionally, this extraction of semantic relations has focused on pairwise protein–protein interactions (PPI). Proteins are the macromolecules that make up most biological structures, and PPIs are a central aspect in the metabolism of all living organisms. Annotation and study of large scale protein–protein interaction networks is thus a key method in understanding the role and function of the various proteins encoded by the genome.

Text mining for these PPIs has proceeded as a task of relation extraction, where mentions of protein names are detected in natural language text, and PPIs are defined as the pairs of these protein mentions that the text states as interacting. The advantage of such pairwise PPIs is a straightforward annotation scheme and an extraction target easily approached by e.g. classifier-based machine learning methods. However, such PPIs can capture only a limited subset of the relations described in literature, or capture the relations at a very low granularity.

Event extraction has been proposed as a more detailed alternative for PPI detection. Events are complex relations that aim to provide an annotation scheme detailed and flexible enough to fully capture the semantics of natural language with a formal representation. Events aim to describe not only which concepts in the text are stated to be interacting, but what the types of those interactions are, and what roles the concepts have in them. Extraction of such events aims to produce a formal representation for any text, a set of facts that can bring the knowledge within the scientific literature to use in computational applications supporting research and medicine.

Events in BioNLP are generally considered to have the following characteristics: 1) their arguments have a type and a direction, 2) they have annotated trigger words (often verbs) and 3) they can be nested, that is, an event can act as an argument of another event (See Figure 1.1).

The development of event extraction for BioNLP has largely been driven by corpus annotation, most notably the GENIA project (Kim et al., 2008). An earlier effort was the BioInfer corpus, which developed a complex, nested annotation scheme (Pyysalo et al., 2007). The real breakthrough of event extraction in the BioNLP field came from the BioNLP Shared Tasks, a competitive evaluation of event extraction software, organized by the GENIA project in 2009, 2011 and 2013 (Kim et al., 2009, 2011a; Nédellec et al.,



Proteins	Triggers	Relations	Events
P1 = "STAT3"	T1 = "phosphorylation"	R1 = (P1, P2)	E1 = Phosphorylation<T1>(Theme:P1)
P2 = "Vav"	T2 = "involve"	R2 = (P1, P3)	E2 = Regulation<T2>(Cause:P2, Theme:E1)
P3 = "Rac-1"			E3 = Regulation<T2>(Cause:P3, Theme:E1)

Figure 1.1: Events and relations. A GENIA-style event annotation is shown above the sentence and the corresponding binary relation annotation below the sentence. These traditional binary relations consist of untyped, undirected pairs of interacting proteins. The events consist of a trigger and directed arguments. In the *Regulation* events, the order of the *Theme* and *Cause* arguments defines the direction of the event. The *Phosphorylation* event E1 is nested as an argument in the two *Regulation* events. While more complex than the relations, the events describe the semantics of the sentence more accurately.

2013). These shared tasks popularized the event extraction approach, and provided a clearly defined, competitive setting for the development of automated methods for biomedical event extraction.

1.2 Research Questions

With biomedical event corpora having become available for research, the primary research question addressed in this thesis is, what sort of approaches can be developed for automated extraction of these complex semantic structures?

Very early in the work it became apparent that the complexity of the natural language favors the use of machine learning over manual definition of rules for event extraction. Following this choice of approach, the research questions addressing the practical formulation of a system for event extraction are two-fold: First, addressed in Chapter 2, how can the extraction of complex event structures be defined as a task approachable by supervised machine learning techniques? Second, addressed in Chapter 3, how can the information available in the text, such as syntax, be utilized effectively in machine learning of events?

Only after the successful development of an event extraction system do practical applications become a possibility. With the large scale of literature available, the final research question, addressed in Chapter 4, examines how the developed methods can be applied to real-world event extraction tasks with the aim of producing information useful in biomedical research.

In the rest of the current chapter we see how these research questions evolved over time and were approached during the course of this work, and how the solutions developed became the Turku Event Extraction System (TEES), a comprehensive automated approach for the task of computational event extraction.

1.3 Development of the Turku Event Extraction System

The University of Turku established a project for the study of biomedical natural language processing in 2002. The aims of this project are to develop methods and resources for text mining by analyzing the structure and attributes of biomedical text, producing annotated corpora and developing automated systems for information extraction.

The most widely cited project of the Turku BioNLP research is the BioInfer corpus, which was the first detailed, "event type" annotation developed for biomolecular interactions (Pyysalo et al., 2007). The much larger GENIA event corpus was published shortly thereafter (Kim et al., 2008), and

these resources allowed the Turku group to start working on developing complex relation mining tools from the year 2008. These efforts included an implementation (Pyysalo et al., 2008) of the RelEx system of Fundel et al. (2007), and a graph kernel developed by Airola et al. (2008a). However, these systems were still mostly targeted for extraction of PPI-type binary relations. A *binarized* version of the BioInfer corpus was also developed to fit the more detailed annotation into the existing PPI paradigm (Heimonen et al., 2008).

Work on what was to become the Turku Event Extraction System began in Autumn 2008, based on a study analyzing the similarity of syntactic parse trees and event-type complex semantic relation graphs (Paper I). The study illustrated a strong correlation between complex event annotations and collapsed Stanford dependency parses (a syntactic parse scheme developed for semantic information extraction). It was shown that for words linked by an event annotation, corresponding syntactic dependencies usually existed, connecting the words through only a few dependencies (or just one), even when these words were distant in the linear order of the sentence. As existing tools could already automatically generate dependency parses, parsing formed a logical starting point for the development of automated event extraction systems.

First attempts at utilizing dependency parses involved defining a set of syntactic patterns that would allow automated detection of corresponding semantic relations. However, due to the diversity of syntactic structures, this rule-based approach was quickly abandoned in favor of machine learning. Taking cues from the graph kernel approach, a feature based system was developed, using the concept of the *shortest path*, the smallest set of dependencies connecting two words for which a semantic relation may exist. The diversity of such syntactic structures was addressed by chopping them down into smaller atomic components such as syntactic n-grams. These features were then used to train a classifier, a machine learning system that would extract the real semantic relations from all the potential candidates.

To address the whole task of extracting events from natural language, a simple approach was devised also for detecting *trigger words*, keywords such as the verb "phosphorylates" in "IKK phosphorylates IκBα". These words, explicitly marked in event annotations, define the interactions between gene and protein named entities, and their detection can be viewed as a task similar to named entity recognition.

To approach trigger word detection with similar machine learning techniques as used for relation detection, we defined this problem as a classification task where each trigger word entity was reduced to a single syntactic parse token, the *head token*, enabling trigger detection by simply classifying all tokens in the text. In working form by the end of autumn 2008, these classification methods formed a generalized technique for detection of

event-type semantic relations using a graph representation, and made up the basis for what would become the two-step trigger/argument detection approach used in the Turku Event Extraction System (TEES). The semantic relation detection system was introduced at the NODALIDA'09 conference, becoming public only shortly before the 2009 BioNLP Shared Task workshop (Björne et al., 2009a).

1.4 BioNLP'09 Shared Task on Event Extraction

Shared Tasks in computer science and related fields are competitions where the participants are given a common goal, and within a certain timeframe have to produce a solution to the defined problem. In information extraction, the participants can be given a certain dataset on which to develop their systems, and at the end of the competition, the systems are evaluated by the organizers using another dataset hidden from the competitors. The BioNLP Shared Task is a competition in the BioNLP field, concerning the automated extraction of events from different domain corpora. The first BioNLP Shared Task was organized by the University of Tokyo in 2009 (Kim et al., 2009), with 24 participating university teams submitting final results.

Based on our ongoing work on relation extraction, the first version of the Turku Event Extraction System was developed by the University of Turku team for the BioNLP 2009 Shared Task (Paper II). While the existing relation and entity detection systems provided a basic framework on which to build, the BioNLP Shared Task corpus, based on the GENIA corpus, was a new and different extraction target, and much work was done to build a system suitable for addressing this task. In particular, the original graph-based approach could only detect one event per trigger word, leading to the development of an *unmerging* system for detection of overlapping but separate events (such as the two *Regulation* events shown in Figure 1.1).

Much to the happy surprise of everyone involved, the Turku Event Extraction System performed remarkably well, reaching first place and being the only system with performance above 50% F-score. In retrospect, good knowledge on the capabilities and limitations of the machine learning systems available for the team, leading to careful optimization and tuning of parameters, was likely a key aspect in the high final result. Following the Shared Task, TEES was extended to predict subtasks 2 and 3, the parts of the BioNLP'09 Shared Task concerned with subprotein interactors (e.g. domains) and modality detection (speculation and negation). Having extended TEES for the full scope of the BioNLP Shared Task event definition, the next goal was to evaluate the suitability of this event scheme for real-world information extraction.

1.5 PubMed-scale Event Extraction

The BioNLP'09 Shared Task provided preannotated named entities, as well as automatically generated syntactic parses, so the participants were free to focus on the event extraction itself. However, the resulting systems could not be used alone for detecting events from unannotated text, and choosing a set of additional tools for doing this was the first step in applying event extraction to unannotated text. Based on experiences from the Shared Task, the Charniak-Johnson parser (known today as the BLLIP parser), using the biodomain-adapted McClosky model, with conversion to dependency format using the Stanford parser tools, was chosen for producing the syntactic parses (McClosky and Charniak, 2008; McClosky, 2009; de Marneffe and Manning, 2008). For detecting named entities, the protein and gene names that are the arguments of the events, the BANNER system (Leaman and Gonzalez, 2008) was used. Unprocessed text was split into individual sentences with the GENIA Sentence Splitter (Kazama and Tsujii, 2003a).

For the dataset, all PubMed abstracts, available for download from NLM, were chosen. At the time, processing all of PubMed was not yet quite commonplace, and with the computational resources provided by CSC, we felt that analyzing such a large dataset would prove a definitive analysis of the capabilities of event extraction, even if it would take a lot of computational resources. Even with CSC cluster computers, working through all of the text in the PubMed abstracts would take a while. Analysing the data would also produce a lot of material, so some of the work was divided into a preliminary publication on the results for 1% of the whole dataset (Paper III). This work was finished at the very end of 2009, aimed for publication at the ISMB 2010 conference. Several analyses were performed on the randomly selected 1% sample, including our first attempt at constructing an event network by joining events through string matching of BANNER-detected protein names.

The work on the whole PubMed abstracts dataset progressed much faster than initially predicted, and the final dataset was ready already at the end of spring 2010, leading to the corresponding publication at the BioNLP workshop in July 2010, just a few days after the 1% analysis became public (Björne et al., 2010b). Complementing each other, these two publications aimed to provide a comprehensive overview of the event dataset extracted from PubMed abstracts, by both large-scale quantitative, as well as detailed qualitative measurements.

The PubMed abstracts dataset, based on the NLM 2009 release, contained nearly 18 million citations. In total, 19 million biomedical events were extracted from it using TEES. A more effective event normalization approach was developed, aligning similar protein or gene names with e.g. prefix and suffix removal, and aligning events sharing the same structure as

well as the same normalized protein and gene names. As a result, the 19 million event instances were normalized into 4.5 million unique events.

With this more advanced normalization, network construction was taken a step further, in an experiment to re-create a KEGG signaling pathway. The human apoptosis pathway was chosen for its relative complexity, its importance in cellular signaling and the familiarity of the researchers with the related biochemistry. Taking the protein nodes as given data from the KEGG network, the goal was to reconstruct the interactions connecting them. The experiment showed that all of the interactions could be recovered from the text mining data, often also with correct interaction types. However, the experiment also showed that an event is not conceptually the same thing as a physical, biomolecular interaction, as it is common to state that one protein regulates another (especially if these are well known, central proteins) even when the actual interaction happens through several intermediate proteins. While signaling networks denote the direct biomolecular interactions linking proteins together, text mined events can, in addition to these direct interactions, refer to higher-level conceptual relations. Whether this conceptual flexibility will prove to be troublesome noise, or an additional source of new insights, remains an open question.

1.6 BioNLP'11 Shared Task

After completing the PubMed-scale event extraction work, the next iteration of the BioNLP Shared Task, BioNLP'11, started in the autumn. The BioNLP'11 Shared Task was a significantly expanded effort, having multiple organizing teams from several universities in addition to the University of Tokyo. The goal of the shared task was to extend the BioNLP'09 event scheme to various new domains. While the basic scheme from the BioNLP'09 task already allowed for rather extensive coverage of basic biomolecular interactions, the event approach held potential for far more diverse applications. Thus, the BioNLP'11 Shared Task had eight different tasks, ranging from epigenetics to protein structural relations and syntactic co-reference to host-pathogen interactions.

In the intervening time since the BioNLP'09 Shared Task, further developments had happened in the field of biomedical event extraction. Most notably, the EventMine system developed by Miwa et al. (2010), following the basic extraction approach of TEES, achieved a new highest performance of 56.00%. Besides extensive work on system optimization, the EventMine performance relied on the use of combined parses, producing combined features from the output of several automated syntactic parsers with presumably different areas of strength. As parse combination is a very time consuming process, both computationally and in terms of system development

time, this strategy was not considered an option in extending TEES for the BioNLP'11 Shared Task. TEES performance was by now lagging behind the highest known results, and as further gains from optimizing the basic system were an unrealiable goal at best, the goal of extending TEES for the BioNLP'11 Shared Task was set as generalization, instead of competing on absolute performance. TEES was to participate in every single task and subtask of the BioNLP'11 Shared Task, ideally with minimal task-specific adaptation, to demonstrate the generalizability of the graph-based approach to event extraction (Paper IV).

The graph-scheme of representing named entities and trigger words as nodes, and event arguments and relations as edges, fortunately proved to be applicable to all eight BioNLP'11 domain tasks. As the graph-scheme had its roots in binary protein-protein interaction extraction it was also directly applicable to the triggerless events and relations introduced in some BioNLP'11 tasks. All in all, no new compromises needed to be made to convert the BioNLP'11 dataset to the "Interaction XML" format used by TEES, and the full shared task annotation could be preserved, apart from a few instances where mapping of protein domain arguments remained slightly ambiguous, as was the case in the BioNLP'09 Shared Task, too.

TEES had been open-sourced and published as the 1.0 version the preceding spring, and thus the quickly written experimental code had been refactored to a more generalized form. Extending the entity node and interaction edge detection modules was relatively straightforward, and use of object oriented Python kept the complexity of the task at a manageable level. In autumn 2010, TEES participated in the Coreference (CO), Entity Relations (REL) and Bacteria Gene Renaming (REN) tasks, the three "supporting tasks" of the BioNLP'11 Shared Task, reaching first place in the REL and REN tasks. As new system features, TEES for the first time took advantage of external structured knowledge to great effect in the REN task, in the form of known renaming pairs derived from the Uniprot *B. Subtilis* gene list and the *B. Subtilis* research community wiki, the *Subti*Wiki.

The BioNLP'11 supporting task phase ended in 2010, and in spring 2011 the five main tasks became available for the competitors. The tasks were much larger, and the schedule perhaps even tighter than on the supporting tasks. Now more than ever, the unified graph approach and the modular design of TEES proved vital for completing the given tasks on time. The same basic event extraction approach was used for all tasks, quickly providing automated prediction systems for all of the varied domains. Building on these results, time could be applied to task specific optimizations in the form of additional program modules. Time was however limited, and especially the costly parameter grid search, tuning all individual components' parameters against the final metric, meant that not very much iterative development was possible on the tasks with the larger datasets. Ultimately, TEES achieved

the first rank in two of the five main tasks, and stayed close to the best systems on the GE task, the direct continuation of the BioNLP'09 Shared Task.

1.7 DDIExtraction'11 Shared Task

Shortly after the BioNLP'11 results were in, a new Shared Task was announced. The "First Challenge task: Drug-drug Interaction Extraction", organized by the Universidad Carlos III de Madrid, presented a text mining task that was traditional in its PPI-like binary relation annotation scheme but highly novel in its important medical subject domain: the detection of adverse patient reactions caused by unforeseen interactions between co-administered drugs (Segura-Bedmar et al., 2011). Such drug–drug interactions can be lethal to patients, and up-to-date information may not be easily available for doctors prescribing the drugs, meaning text mining could help in keeping up with the individual reports in medical journals hinting at such dangers. The clarity of the annotation scheme and the ease of use of the datasets probably contributed to the high participation in the task despite the tight schedule. All in all, 10 teams from various universities participated in the DDIExtraction'11 shared task. In the final results, TEES placed 4th, with performance at 96% of the highest performing system (Paper V).

From the point of view of biomedical event extraction the DDIExtraction'11 shared task was very interesting, as it provided an opportunity to compare an event extraction system to binary relation extraction tools in the post BioNLP Shared Task text mining field. Since TEES has no direct concept of events (representing them as a graph of trigger nodes and argument edges) and uses the same graph scheme also for binary relations (with individual relation edges connecting named entities), applying the system to the DDIExtraction'11 task was very straightforward. The task dataset was also distributed in a format similar to the "Interaction XML" used by TEES, allowing even faster development. Linking TEES back to our earlier work on binary relation extraction, we applied the RLS machine learning system, known from the graph kernel project to work well on this type of data. We also used "thresholding", optimizing the positive/negative classification cutoff, a technique that proved very effective (increasing performance by over 6 percentage points) but which can only be easily applied on binary classification tasks. Following experiences from the BioNLP'11 Shared Task, TEES again made use of external datasets in the DDI task, and this paid off in two percentage points of additional performance.

Several systems participating in the task utilized the large scope of research available on kernel methods developed for binary relation extraction. The graph-kernel developed at University of Turku by Airola et al. (2008b) was used by several teams, including the winning one.

1.8 Publishing the Turku Event Extraction System

Following the DDIExtraction'11 task, TEES had participated in three global Shared Tasks. The initial 1.0 release, made available after the original BioNLP'09 Shared Task, was largely outdated by now. Extending TEES to the eight domain tasks of the BioNLP'11 Shared Task, as well as the new domain of the DDIExtraction'11 task, had produced a mess of a system, with lots of special case code and task specific data merged inside the program. As demonstrated by the high rankings in the Shared Task, the system would clearly be useful for research on biomedical text mining, but it could only be so, if it was actually usable. From late 2011 until August 2012, a major re-engineering effort was made, finally producing the 2.0 release of TEES. The original system had relied on "pipeline files", effectively long procedures of Python-code to call the required components of each experiment. While very flexible, the complexity of the system had grown beyond this approach. To make TEES again useful for end-users, two main approaches guided the refactoring. First, a higher-level object oriented interface was developed, encapsulating all of the feature generation, machine learning and parameter optimization code. Second, a model file encapsulating all the datafiles that form the model learned for a specific task was developed. The updated TEES 2.0 was published as a full open source project on GitHub. In Section 4.5 the importance and results of making research code publicly available are discussed.

The PubMed-scale event extraction work has also been continued, and the new extraction targets that TEES was adapted for in the BioNLP'11 Shared Task have been applied on PubMed-scale datasets. To make the extracted mass of events usable for actual applications, the EVEX project was founded to build a fast database structure for handling the data, for connecting the text mining results to biomedical databases via normalization and for managing real-world use cases on biological collaborative projects (Van Landeghem et al., 2011). The first version of EVEX was published in 2011, and the development of the database and web interface has been ongoing ever since. TEES has been used for this work first in producing the base GENIA-style events, and in early 2012 also epigenetics events and protein structural relations, for all PubMed abstracts and PubMed Central full text articles. To simplify these real world applications, in the 2.0 version TEES was also extended with a pipeline of open-source, high-performance NLP tools for syntactic parsing and other pre-processing tasks.

Connected only by the extensible "interaction XML" file format, these Python-based tool-wrappers automate the setup and use of varied NLP tools, simplifying the construction of a text mining pipeline. With the help

of this toolchain, TEES 2.0 can finally process fully unannotated text, e.g. receiving an abstract from PubMed or a text file, and automatically producing a semantic annotation for any of the 10 included biomedical domain targets.

1.9 BioNLP'13 and DDIExtraction'13 Shared Tasks

The BioNLP and DDIExtraction shared tasks were organized again in 2013, and TEES participated in both of them, with the aim to continue the approach of generalization introduced in TEES 2.0. TEES 2.1 introduces a system for automated annotation scheme learning, allowing TEES to be directly applied to new annotations and new corpora, whereas in earlier versions the constraints of each annotation scheme had to be manually defined in program code.

With this approach, TEES could be applied with almost no task-specific optimization to all BioNLP'13 tasks. As with the 2011 task, TEES 2.1 reached a first place in four out of eight tasks (Paper VI). In the DDIExtraction 2013 task, TEES 2.1 reached 2nd and 3rd places in the drug NER and drug–drug interaction tasks, respectively (Björne et al., 2013). The performance of TEES in the 2009–2013 shared tasks is summarized in Table 1.1.

To make the use of TEES easier for other task participants, we also published our predictions during the system development phase of these competitions, so that other teams could build on them. In the BioNLP'13 Shared Task, the best result for the GENIA task was achieved by a system utilizing TEES predictions, and in the DDIExtraction'13 task the best performing system for the drug–drug interaction task included TEES among several systems in a system combination approach (Hakala et al., 2013; Thomas et al., 2013). Following the results of the BioNLP'09 Shared Task (Kim et al., 2009), these results indicate that by combining different text mining systems, it is possible to utilize the strenghts of different approaches and reach better overall performance.

1.10 Disposition of the Thesis

The primary contribution of this thesis, the six peer-reviewed research publications form the latter part of the book. Papers II, IV and VI cover the three BioNLP Shared Tasks, describing the most important milestones in the development of the Turku Event Extraction System. Paper I introduces the dependency parse analysis that formed the starting point for developing the system, Paper III demonstrates the applicability of TEES to real-world text-mining tasks and Paper V provides a comparison with binary PPI extraction.

Task	Name	Rank	#	Score	Δ
GE09 [1]	GENIA Event Extraction	1	24	51.95	5.29
GE09 [2]	Protein Site Arguments	–	6	–	–
GE09 [3]	Negation & Speculation	–	6	–	–
GE11 [1]	GENIA Event Extraction	3	14	53.30	-2.74
GE11 [2]	Protein Site Arguments	2	4	41.98	-3.88
GE11 [3]	Negation & Speculation	1	2	26.86	0.03
EPI11	Epigenetics and PTM:s	1	7	53.33	18.3
ID11	Infectious Diseases	5	7	42.57	-13.02
BB11	Bacteria Biotopes	3	3	26	-19
BI11	Bacteria Gene Interactions	1	1	77	–
CO11	Protein/Gene Coreference	4	6	23.77	-10.28
REL11	Entity Relations	1	4	57.7	16.1
REN11	Bacteria Gene Renaming	1	3	87.0	22.6
DDI11	Drug–Drug Interactions	4	10	62.99	-2.75
DDI13 9.1	Drug Name Recognition	2	6	60.4	-4.8
DDI13 9.2	Drug–Drug Interactions	3	8	58.7	-6.1
GE13	GENIA Event Extraction	2	10	50.74	-0.23
CG13	Cancer Genetics	1	6	55.41	3.32
PC13	Pathway Curation	2	2	51.10	-1.74
GRO13	Gene Regulation Ontology	1	1	21.50	–
GRN13	Gene Regulation Network	3	5	0.86 SER	+0.13
BB13 [1]	NER and Categorization	–	4	–	–
BB13 [2]	Bacteria Localization	1	4	42	2
BB13 [3]	Bacteria Entities & Relations	1	2	14	8

Table 1.1: Turku Event Extraction System in the shared tasks. The ranking and score of TEES in the tasks and subtasks of BioNLP Shared Task 2009, 2011 and 2013, as well as DDIExtraction tasks 2011 and 2013 are shown. Subtasks are indicated with *[n]*. The number of participants is indicated by column #. The *Score* and Δ are in F-score except for task GRN13 where SER (Slot Error Rate, smaller is better) was the primary metric. The Δ shows the performance difference to the 2nd ranked system (when *Rank* is 1) or to the 1st system (when $Rank > 1$).

The first part of the thesis, chapters 1–6, draws together and analyzes the common thread in the publications, the development of the Turku Event Extraction System. Chapter 2 introduces the main research question of the work: developing a machine-learning approach for biomedical event extraction. Chapter 3 gives a detailed overview of the Turku Event Extraction System and the text mining techniques used for event extraction. Chapter 4 describes the application of event extraction to a real-world text mining task, in the context of the PubMed-scale event extraction project. Chapter 5 is the overview of related work, in both the event and PPI extraction fields, and Chapter 6 concludes the first part of the thesis with an analysis

of the contributions of the work done, as well as some thoughts on future directions for biomedical event extraction.

Chapter 2

Event Extraction as a Machine Learning Task

Machine learning has been used in the Turku Event Extraction System starting from the earliest prototypes. The most important reason for this choice was the positive experiences from the *graph kernel* project which used a classifier for binary PPI detection (Airola et al., 2008b). Analysis of the shortest dependency paths corresponding to event argument edges revealed a considerable variety of syntactic structures, further empasizing the suitability of machine learning over rule-based approaches (Paper I).

This early choice to base TEES on machine learning would have several consequences on both the approach to event extraction as well as the design of overall experimental strategy. In particular, using a classifier led to an "atomic" approach to event extraction where the task of predicting complex event structures was decomposed into sequential, independent classification tasks (Paper II). In the later Shared Tasks, machine learning was a key component in enabling TEES to be rapibly adapted to all the varied domain tasks introduced in that competition (Papers IV, V and VI).

In this chapter, we first see how the complex graph generation task of event extraction can be defined as a set of individual classification tasks. The external classifier programs used by TEES are introduced, as well as the performance measures specific for event extraction. Finally, solutions developed for optimizing the multi-step classification pipeline of TEES are described in detail.

2.1 Defining Event Extraction as a Classification Task

Biomedical events, such as those defined in the GENIA corpus, are complex nested structures, displaying a large amount of variation in order to capture

all possible statements of interest in natural language text. Creating such structured output with machine learning systems can be a difficult task, and as we'll discuss later, several alternative methods have been proposed for this problem. One of the most straightforward approaches, and the one used by TEES, is to convert the structured prediction problem into a number of independent (binary or multiclass) classification tasks.

The starting point in the construction of the Turku Event Extraction System was an analysis of the similarities between dependency parse and complex semantic annotations, such as those in the BioInfer and GENIA corpora (Paper I). Both of these annotations could be modelled as directed graphs (mostly acyclic), so this comparison was essentially an analysis of graph similarity.

The "Interaction XML" file format used in TEES, developed originally for unifying the annotation schemes of five binary interaction corpora (Pyysalo et al., 2008), represented semantic annotations as named entity nodes connected by interaction edges. The graph kernel machine learning system also used this format, and depicted the dependency parse as a directed graph, used in the construction of the graph kernel's adjacency matrix (Airola et al., 2008b).

TEES development started as the search for a method for extracting the complex semantic relationships defined in the BioInfer and GENIA corpora. In light of the mentioned earlier work, development of TEES was based on a perspective of viewing both the parse and the semantic annotations as graphs. Compared to binary interactions, the complex relationships and events defined in the BioInfer and GENIA corpora are larger structures, usually consisting of a trigger node and its arguments. Like the named entities (protein and gene names etc), triggers are mapped to defined spans of text (often verbs). The arguments of these structures have type and direction, and can be considered a form of binary relation, linking not only named entities, but also the trigger nodes. Atomized in this way, the nested event structure becomes a graph of trigger and named entity nodes, connected by event argument edges.

Prediction of this graph for a single sentence thus becomes a task of predicting the trigger (and sometimes also named entity) nodes, and predicting argument edges linking them. If trigger nodes are limited to one per syntactic token, such a graph can be generated in two classification steps. First, each syntactic token is classified as either a negative, or one of a number of trigger classes (See Figure 2.1 D). Second, each valid directed pair of (trigger or named entity) nodes is classified as either a negative, or one of a number of event argument classes (See Figure 2.1 E). These two multi-class classification steps produce an interaction network that can represent in a merged form the full event annotation for a sentence.

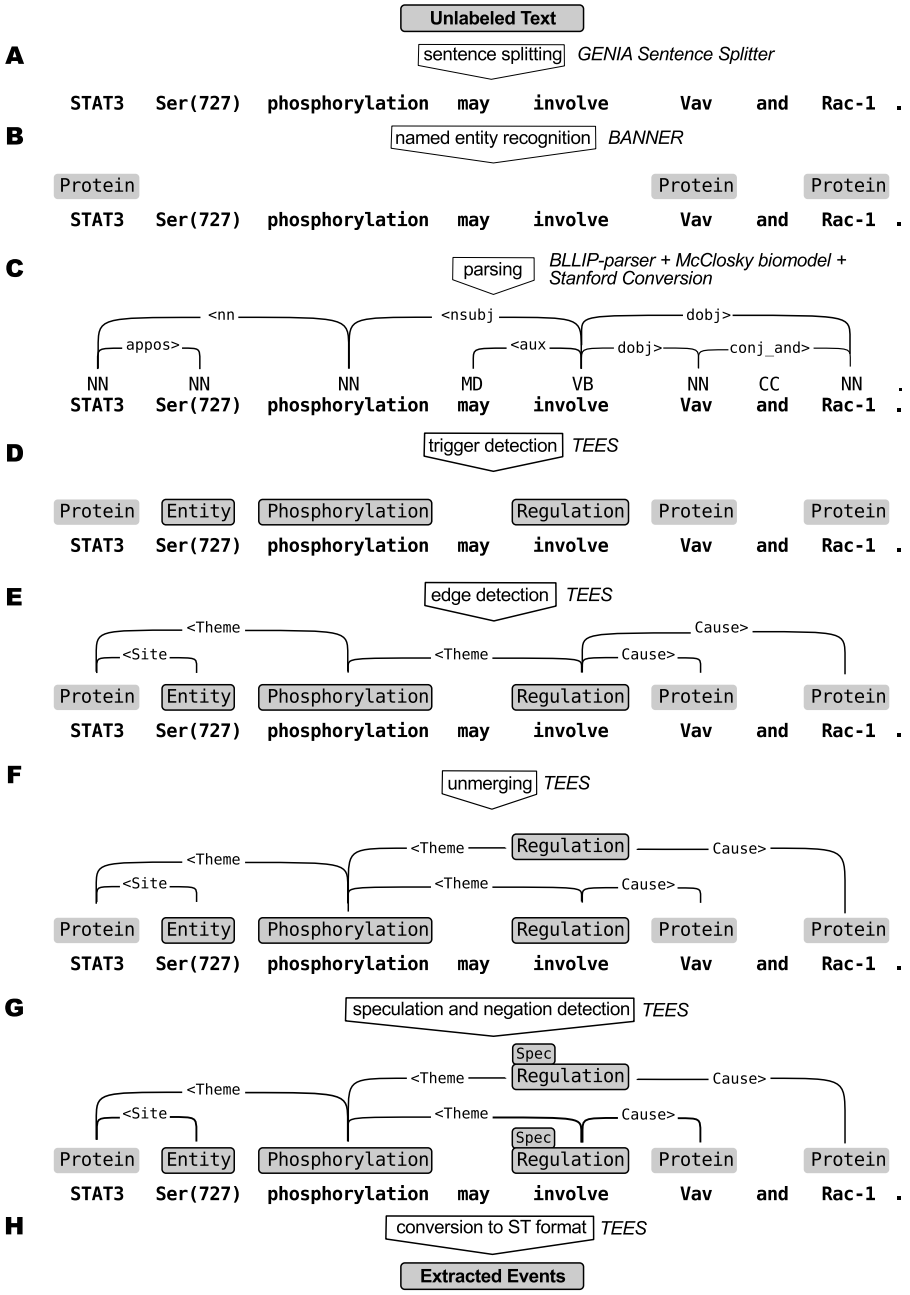


Figure 2.1: Event extraction. After conversion to ASCII, text is split into sentences (A), where gene/protein entities are detected (B). Each sentence with at least one entity is then parsed (C). Events are extracted from the parsed sentences (D-G). Figure adapted from Paper IV.

Such a simplified view of events was the only one used in earlier versions of TEES (Björne et al., 2009a). However, with the BioNLP'09 Shared Task, it became necessary to address the prediction of event structures in full detail. The limitation of the graph structure is demonstrated by Figure 2.1 E. If only one trigger can exist for each word token, events with the same trigger, but different arguments become merged together. However, at the time of trigger detection, we don't know yet how many "copies" of that trigger are needed, as event arguments are not yet defined. After argument edges are predicted for each valid node pair, it becomes possible to "unmerge" the overlapping events. In the event annotation scheme, the type and number of arguments for each event is limited by the type of its trigger word. For example, a *Regulation* type event must have at least one *theme* argument and can optionally also have one *cause* argument. Using this information, the graph is "pulled apart" by duplicating trigger nodes with invalid argument combinations, and dividing the arguments into valid combinations among the new nodes (Figure 2.1 F). This step can also be represented as a binary classification task, with each potential trigger and argument combination being classified as either a true event or a negative.

The result of these three consecutive classification tasks is an event network where full, valid events are defined by a trigger node and its set of outgoing edges. Finally, event modifiers such as *speculation* and *negation* can be predicted simply by classifying each event as positive or negative for each potential modification type (Figure 2.1 G).

Thus, the complex task of predicting structured output in the form of events has been broken down into a set of consecutive classification tasks. All of these tasks are separate machine learning problems, requiring independent training of classifiers, and employing a large amount of different features derived from the text.

2.2 Classifiers Used in TEES

The earlier work on the graph kernel had relied on the RLScore software which was still at an experimental development stage when the TEES project began, so the high performance SVM-light software of Thorsten Joachims was chosen as the classifier solution (Joachims, 1999). Like RLS, SVMs are known for their ability to handle large datasets, huge amounts of features and noisy data, all common qualities in text mining. In the DDIExtraction'11 Shared Task, RLS was again used alongside an SVM as a classifier (Paper V).

2.2.1 Support Vector Machines

The Support Vector Machine is a set of supervised machine learning algorithms, introduced originally by Cortes and Vapnik (Cortes and Vapnik, 1995). The SVM can be used for both classification and regression, and is a popular method in natural language processing due to its ability to handle both large datasets and noisy data. In TEES, a linear SVM is used for all SVM classifications.

The data classified by a linear SVM is represented as a set of examples, defined by a class (to be predicted) and a set of features (known data on which classification is based). On a general level, the SVM classification is based on defining a hyperplane that best separates the two classes. This hyperplane exists in n-dimensional space, where each feature is represented by its own dimension. In a trivial case, with only one feature, the hyperplane becomes a scalar value, with two features, a line, with three features, a plane, and with more than three features, a hyperplane.

Support vectors are the points closest to the hyperplane, in other words, the most ambiguous examples, which are the hardest to classify and key to defining a hyperplane that produces a good separation. This hyperplane is chosen so that the distance to the nearest example is maximized (See Figure 2.2). This distance between the hyperplane and the closest known example is called the margin. The SVM optimization problem was greatly simplified by Cortes and Vapnik who introduced the concept of the soft margin to better handle mislabeled examples (Cortes and Vapnik, 1995). The original SVM formulation was a linear classifier, but since then non-linear kernels have also been introduced. While non-linear kernels can demonstrate better performance on some machine learning problems, linear SVM:s allow for a number of optimizations that considerably speed up their training, making them more suitable for classifying the large datasets used in TEES.

The original SVM algorithm was developed for binary classification. In this work, we deal mainly with multiclass-classification problems, where the example can belong to one of a number of classes. Multiclass SVMs can be divided into two categories: Those that treat the problem as multiple binary classification problems (e.g. one-versus-all, where each example is classified as belonging to one class, or any of the other classes) and those that attempt to solve the whole task as one optimization problem. In the Turku Event Extraction System, the SVM^{multiclass} (Tsochantaridis et al., 2005) implementation[1] is used for all classification tasks, except where the regularized least squares system RLScore is used as an alternative classifier. SVM^{multiclass} has been optimized for fast linear classification, with the runtime scaling linearly with the number of training examples.

[1] `http://svmlight.joachims.org/svm_multiclass.html`

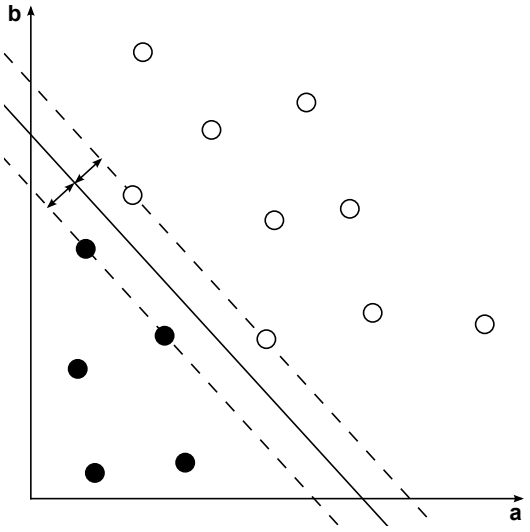


Figure 2.2: The maximum-margin hyperplane for a binary SVM classifier. With two features (a and b) the hyperplane is a line. The hyperplane is chosen to maximize the margins (the dotted lines) and the separation of the two classes (black and white). The four examples on the margins are the support vectors.

2.2.2 Regularized Least Squares

Regularized least squares classification (RLS), also known as least squares SVM, or ridge regression, is the second classification technique used in this work (Rifkin et al., 2003; Evgeniou et al., 2000). It is closely related to SVM classification and often displays similar performance, but has the advantage of efficient computational shortcuts for fast cross-validation evaluations.

Where training SVMs involves solving a convex quadratic program, training an RLS classifier requires the solution of a single system of linear equations. To achieve good computational performance, RLS implementations rely on multiple approximations.

The RLS implementation used in this work is the RLScore[2] software package, developed at the University of Turku, and earlier utilized in a text-mining setting in Airola et al. (2008b). In this work, RLScore is used in Paper V for classification of drug–drug interactions.

2.3 Performance Measures

For optimizing a machine learning system, the choice of the performance metric is critically important. In TEES development the micro-averaged

[2] `http://www.tucs.fi/RLScore/`

F-score has been most widely used. The F-score (also known as F_1-score or F-measure) is the harmonic mean of precision (fraction of correct predictions) and recall (fraction of correctly extracted positives). In the following formulas, TP, FP, and FN refer to true positives, false positives and false negatives.

$$precision = \frac{TP}{TP + FP} \quad recall = \frac{TP}{TP + FN} \tag{2.1}$$

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{2.2}$$

The F-score is a binary measure, but with some modifications it can also be used in a multi-class setting. In such settings, the per-class F-scores are combined for an overall measure, usually by macro- or micro-averaging (Sebastiani, 2002). In micro-averaging, individual decisions are summed over all classes.

$$precision = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FP_i} \quad recall = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FN_i} \tag{2.3}$$

In macro-averaging precision and recall are evaluated per-class, then averaged over the classes:

$$precision = \frac{\sum_{i=1}^{|C|} precision_i}{|C|} \quad recall = \frac{\sum_{i=1}^{|C|} recall_i}{|C|} \tag{2.4}$$

Thus, macro-averaging puts more emphasis on good performance for all classes, even if they are very small, whereas micro-averaging provides a measure of overall performance, if the goal is to maximize the number of correct predictions regardless of class.

In both of these cases the F-score is calculated from the averaged precision and recall with Equation 2.2. These averages are directly usable in multiclass classification tasks where the question is only to assign one of the known labels for each example. For instance, if predicting the part-of-speech label (*noun*, *verb*, *adjective* etc.) of a word token we might not know which of these classes is correct, but we know that for each word token there must exist a part-of-speech label.

However, in TEES event extraction, the task is not only to determine which of several labels an example has, but whether it has a label at all. For example, in trigger word detection, each word token may be a trigger word of a class such as *Binding*, *Phosphorylation* or *Regulation*, but it may as well not be a trigger word at all. In the technical implementation of the system, this "lack of class" is represented simply as an additional class, *negative*,

which may be predicted for an example just like any of the other, positive classes.

When evaluating performance, if this "negative class" was treated like the other classes, it would dominate the micro-averaged F-score. Negative instances form the majority in all TEES classification tasks and a very high F-score would be trivially achieved by maximizing performance on the "negative class", by predicting nothing at all, i.e. everything as *negative*.

Therefore, while the "negative class" is a class in the technical implementation of the classification system, when measuring performance, a predicted negative is considered to denote a *lack of predicted class*. In determining micro-averaged precision and recall, the negative class (index 1) is therefore skipped, resulting in a weighted average of the performance across the positive classes:

$$precision = \frac{\sum_{i=2}^{|C|} TP_i}{\sum_{i=2}^{|C|} TP_i + FP_i} \quad recall = \frac{\sum_{i=2}^{|C|} TP_i}{\sum_{i=2}^{|C|} TP_i + FN_i} \tag{2.5}$$

This micro-averaged F-score which ignores the *negative* class is the main internal performance metric used in TEES. Despite more advanced metrics being available the F-score has preserved its position as the primary performance metric due to its comprehensibility, easy application to multi-class situations and similar measures being used as the official metrics in the shared tasks TEES has been developed for. In the BioNLP shared tasks the official *approximate span matching and approximate recursive matching F-score* has been closely approximated by this internal metric, evaluating events defined in the predicted graph as trigger nodes combined with their outgoing edges.

The final evaluation of published TEES results has always been performed using the official evaluation systems of the relevant shared tasks. These performance measures, usually F-scores, ensure a truly objective measure of the system's performance. The internal micro-averaged F-score has been used as a final metric only in the first publication on a preliminary TEES system by Björne et al. (2009b). During the writing of this thesis we found that the metric as described in this paper is in error, as while it does count false positives (FP_i where i is the predicted class and $>= 2$), it does not count them as false negatives for the true class when the true class is also positive, but different (FN_i where i is the true class and $>= 2$). Although not an accurate micro-average, the formula in that paper is however consistent with the program code and the reported experimental results.

The same incorrect micro-averaging formula has since then been used internally in TEES for optimizing all classification tasks, but has now been fixed for the 2.1 release. In practice, for TEES classification tasks this incorrect definition of micro-average has had only a minor impact on performance,

further evidenced by TEES performance in several shared tasks (that use their own evaluation software) being very high but also close to the internal metric.

The F-score is in many ways a problematic measure of performance. As it is the harmonic mean of precision and recall, many different combinations of precision and recall values can produce the same F-score. Thus, to fully understand a result described by an F-score, it is necessary to go back to the component values and note where on the precision/recall plane the F-score is located (See Paper II, Figure 6.). Another central issue with the F-score is its dependence on the class distribution of the dataset. In machine learning, other metrics are commonly used to overcome this limitation. The AUC (Area Under the Receiver operating characteristic curve) is a binary performance measure that is not dependent on the class distribution (Hanley and McNeil, 1982), and has become popular in machine learning work in recent years. Despite the advantages of AUC, the F-score remains a common measure in the BioNLP field and is the official evaluation metric in the BioNLP and DDIExtraction Shared Tasks. To overcome the issue of the F-score being dependent on the class distribution, a dataset can be stratified when it is divided into e.g. training and testing subsets.

Stratification means simply keeping the class distribution constant across all the subsets. While this does not change F-score's dependence on class distribution, it helps to ensure that systems optimized for F-score on the training set exhibit comparable performance on the test set. While stratification can be easily done with binary data, for event datasets where multiple events of different types appear in a single sentence, it can be more difficult to achieve a balanced class distribution. Stratification is complicated also by the fact that when dividing text mining corpora, all sentences from the same document (article, abstract etc) are generally placed into the same subset as a precaution against similar text in both a training and an evaluation set leading to over-fitting. If subsets are selected randomly, the class distribution will naturally be more stratified the larger the dataset is, so with large corpora this issue, along with other problems arising from limited dataset size are mitigated. Regardless, several participants in the BioNLP Shared Tasks have commented on an apparent gap between performance measured on devel and test sets, but whether this is due to stratification issues, or other reasons such as overfitting, remains an open question.

2.4 Optimizing Machine Learning

A key issue in building a machine-learning based system is parameter optimization. For example, the linear SVM classifier is trained using the training data and a regularization parameter, usually denoted as "c". For a

new classification task, the values of c must be determined experimentally. With non-linear kernels, additional parameters are introduced, which also need to be optimized. Classifier parameters are usually optimized using a grid search, with an optimal parameter set defined as providing the highest performance in an area of the search space where the performance no longer changes significantly. Since the SVM needs to be re-trained for each parameter combination, this grid search can be very time consuming. A nested grid can be used, with more values tested in the likely peak area, and such an approach can also be the basis for more advanced algorithms using automated boundary selection and search area refinement (Staelin, 2003).

In TEES, a simple grid search is used, with a user-defined set of parameters. If a non-linear kernel or additional parameters are used, the classifier wrapper automatically generates and tests all the parameter combinations. However, the choice of the actual parameter values and value ranges to test is left to the user. In most experiments the ideal parameter range is found by first running a coarse grid, then focusing on the likely peak area. As SVM parameter spaces can have e.g. local minima, this kind of semi-automated approach with human oversight is considered more reliable, and since most experiments require only a few attempts to determine the optimal parameter range, full automation is not necessary.

However, using fixed parameter values presents a potential issue during feature engineering. Since changing the features of the system can lead to a change in the optimal classifier parameters, it is possible that after changing the features and retraining with the same parameter value grid, the result becomes non-optimal (the new performance peak not being exactly on a grid point) and would incorrectly indicate worse performance from the new features. In practice, a dense enough parameter grid will avoid this issue, and can be observed by multiple grid parameter values in the peak area resulting in only minor performance differences. Still, a fully automated, iteratively refining grid search would probably provide more reliably optimal peak parameter selection for each possible feature set, and is an interesting future direction for the development of TEES.

2.4.1 Optimizing Consecutive Classification Steps

In the TEES event extraction approach, the final event is the product of multiple consecutive classification steps. Edge detection relies on predicted entities, and both edges and entities are used as input by the unmerging step. All of these classification tasks are optimized by searching for the best parameters for that task, but optimal performance for an individual step might not mean optimal performance for the whole task.

For example, the parameters that give the best F-score for trigger detection may not produce the sort of trigger predictions that form the best

basis for the following edge detection step. Edge detection can work better if more triggers are available (even incorrect ones), as the classification of a potential argument edge can reveal information that is not available when predicting triggers in isolation, and as triggers for which no argument edge is predicted are removed afterwards, overall performance can increase.

To maximise overall performance, several approaches have been tested during TEES development. The obvious approach is to perform a multi-dimensional search, testing each parameter combination to determine the best classifier parameters for the overall performance. With several steps this approach can quickly get unwieldy, even if TEES can parallelize the classifier training in a cluster environment. While testing around 10 regularization parameters is often enough to optimize an SVM, and can be done relatively fast, if we have three such steps to optimize (trigger, argument and unmerging detection), a combined 1000 classifiers to train is already getting a bit excessive, cluster or not.

Another issue with the parameter optimization is the metric used to determine classifier performance. The SVM^{multiclass} software used in TEES uses the F-score for its internal performance optimization, and has a tendency to find a balanced precision/recall performance for each regularization parameter tested. However, for example in the case of trigger detection we want to specifically emphasize recall to overproduce triggers. To achieve this, trigger predictions are processed with a *recall adjustment* step. The prediction strength of the negative class is given a multiplier (< 1.0 for more triggers, > 1.0 for less triggers), affecting the relative strength of the positive class predictions. This way we can first find the regularization parameter that provides optimal SVM performance, then "redistribute" this performance so that more predictions become positive, emphasizing recall at the cost of precision.

Determining the recall adjustment parameter becomes another consecutive step to optimize. The resulting system now has four such parameterized steps: trigger detection, recall adjustment, edge detection and unmerging. While a recall adjustment step could also be used between edge detection and unmerging, this has not been tested in practice, as originally TEES unmerging was performed with a rule-based system, and introducing an additional parameter to optimize would further slow down system development. Even four parameters have proved to be unpractical, and in the current TEES approach, only the recall adjustment parameter is optimized in a context larger than its own performance.

In the final system, trigger detection and edge detection are optimized for their independent, local maximum performance. Using the best classifier models from trigger and edge detection, a number of recall adjustment values are tested, with performance evaluated on the edges predicted as a

result of these three consecutive steps. The unmerging system is trained almost entirely outside this system: training examples are generated from gold-standard data extended with self-classified data, intended to produce more negatives for the system to learn on (See Section 3.8), and unmerging classification is optimized in isolation. This simplified approach greatly improves the speed at which new features can be tested, and compared to optimizing the trigger classification, recall adjustment and edge classification parameters globally, incurs only a small performance loss.

2.4.2 TEES Training Setup

A constant issue in machine learning is over-fitting, developing a system that appears to have good classification performance, but the performance of which is actually based on frivolous features which just happen to randomly correlate with the classes. Such situations are usually present on smaller datasets, but even with larger datasets, as the same data is used over time to iteratively optimize a system, the danger of over-fitting still exists. One approach to reduce the likelihood of over-fitting has been the use of cross-validation, with methods such as ten-fold, leave-one-out or five-times-two cross-validation used to divide a dataset into multiple subsets for training and parameter optimization, thus giving a statistical estimate of performance variation with differing datasets. In the University of Turku text mining projects, extensive cross-validation was used e.g. in the development of the graph kernel (Airola et al., 2008b).

In the BioNLP and DDI Shared Tasks, the final evaluation is done on a hidden test set, the annotations of which are not available to the participants during the development of the competing systems (and in the case of the BioNLP tasks, not even later, to allow continued objective evaluation). This means that the chances of overfitting are considerably reduced, and accidental data-leaks by the participants are also effectively prevented. In the BioNLP Shared Task, the provided annotated dataset is also pre-divided into a training dataset and a parameter optimization, or "devel" dataset. In this simple setup, the classifier is trained on the training data, its performance is optimized against the devel data and finally, the hidden test data is classified with the optimal classifier.

While cross-validation probably provides more accurate optimization results, running such experiments is extremely time-consuming. For example, if 10 parameter values are tested, with ten-fold cross-validation we would need to train 100 classifiers. In the graph-kernel experiments the variance observed in ten-fold cross-validation was rather minor, so in TEES a simple training approach, using only a single train, devel and test set is used. However, to maximize the data available for learning, once optimal parameters have been decided against the devel set, the devel set is merged with the

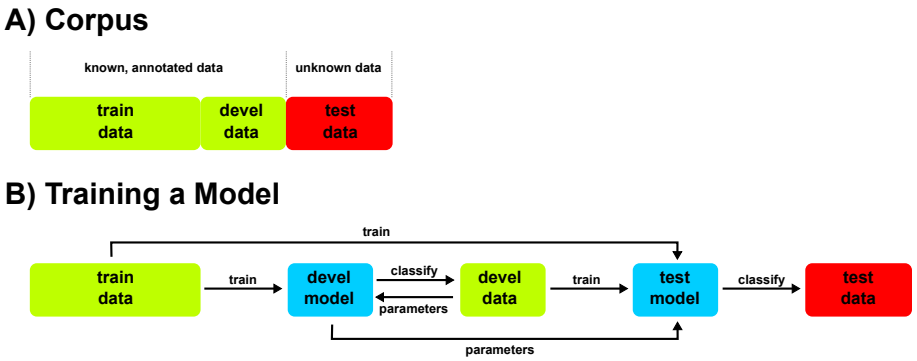


Figure 2.3: TEES training process. A) In the BioNLP Shared Tasks the corpus is divided into annotated *training* and *development* sets, and a *test* set whose annotation is not available to the participants. B) TEES uses this type of three-part division for training an event or relation extraction system. First, the classifier is trained on the training data, and parameters are measured against the development data, producing a *devel model*. For the final classification, a new model is trained using the merged train and devel data, with parameters determined on the devel data, to produce final *test model* which is used to classify the hidden test set. In new event extraction tasks, any corpus can be easily divided into such datasets. When using a fully known corpus, it is important to measure performance against the chosen test set as few times as possible, to avoid over-fitting.

train set, and the whole system is re-trained using this set of all available annotated data with the known optimal parameters. While adding the devel set to the train set can move the optimal parameters away from the values determined on the devel set, this drift is likely to be limited, considering that the devel set should be close to the train set in nature. Thus, in participating in the Shared Tasks, we considered maximizing the available training data to be worth the risk. The TEES training approach is shown in Figure 2.3.

Chapter 3

Syntax Representations for Event Extraction

Having defined an approach for representing event extraction as a set of classification tasks, we now look at how the text itself is presented for the classifier. In this section we review the text mining approaches used in the Turku Event Extraction System, as introduced in Papers II, IV and VI. First, the general principles and techniques for extracting features from natural language texts are introduced in sections 3.1–3.5. Then, the feature representations of the four main classification steps (*entity*, *edge*, *unmerging* and *modifier* detection) are examined in detail in sections 3.6–3.9.

3.1 Syntactic Parsing

Syntactic parsing in computational linguistics concerns the automatic construction of a structure defining the parts of speech in a text, and their syntactic, sometimes also semantic, relations. A syntactic parse is a central source of information in event extraction, and in many systems, including TEES, the most important analysis from which features usable for machine learning are derived.

Parsing is a problem that has been the subject of much research, and with gradual improvements, the performance of syntactic parsers has increased to a very respectable level on many languages. For example, continuous research on parsing the Penn Wall Street Journal treebank has increased performance from 84% to 92% F-score in the years 1995–2006 (Pyysalo, 2008, pp. 66–67).

A multitude of algorithms have been developed for parsing, and with the concurrent advances in machine learning, statistical approaches have become a central building block of modern parsers. Parsing natural language often introduces ambiguous cases not easily solved by a strict formal approach,

but with the "fuzzy" methods of machine learning, an automated system can become robust in handling all the varied structures present in natural language texts.

Several parsing techniques of varying complexity are available and usable for event extraction. While not generally a parsing issue, dividing text into sentences is often a necessary preliminary step in parsing, as parsing software often operates on a single sentence at a time.

For sentence splitting TEES uses the GENIA Sentence Splitter which has been optimized for biomedical texts (Sætre et al., 2007). The GENIA Sentence Splitter uses a maximum entropy method[1] (Kazama and Tsujii, 2003b) to select the actual sentence breaks from all potential candidate break positions (periods, commas etc.). The features used are all based on the text alone (delimiters, previous/next words, special characters etc.) so the input text does not need to be preprosessed in any way. The GENIA Sentence Splitter is reported to achieve an F-score of 99.7% on 200 unseen abstracts from the GENIA corpus it has been trained on.

Another preliminary step required before parsing is tokenization. Parsing considers the roles and relations of the *tokens* that make up the sentence. Generally, each word corresponds to a token, so the simplest form of tokenization is to consider each whitespace-separated word as its own token. However, some additional rules are needed, such as separating punctuation such as commas into their own tokens, or detaching parentheses from the words they enclose. Tokenization is usually done by a parser, but occasionally a better parse can be achieved by applying a separate tokenization method before parsing, for example in the case of domain specific biomedical text. Using a single, shared tokenization can also be helpful when applying several different parsers to the same text in order to produce a combined parse, usable as input in further information extraction tasks.

For individual, tokenized sentences, detailed automated parses can be generated by many different parsing methods. Automated parsing is generally divided into *shallow* and *deep* parsing. Shallow parsing tries to divide the text into its constituents, such as verb or noun groups, but does not define their relation to each other. Deep parsing produces "parse trees", tree or graph structures that connect the words (or word chunks) of the sentence to each other in a hierarchy of linguistic relations such as subject, object, preposition etc.

The first step in generating a parse is POS (part-of-speech) tagging, which means attaching a part-of-speech label (verb, noun etc.) to each token in a sentence. The rule-based Brill-tagger, published in 1992, was one of the first popular English language part-of-speech taggers (Brill, 1992). Different POS taggers use different tag sets, with ones derived from the

[1] `http://www.nactem.ac.uk/tsuruoka/maxent/`

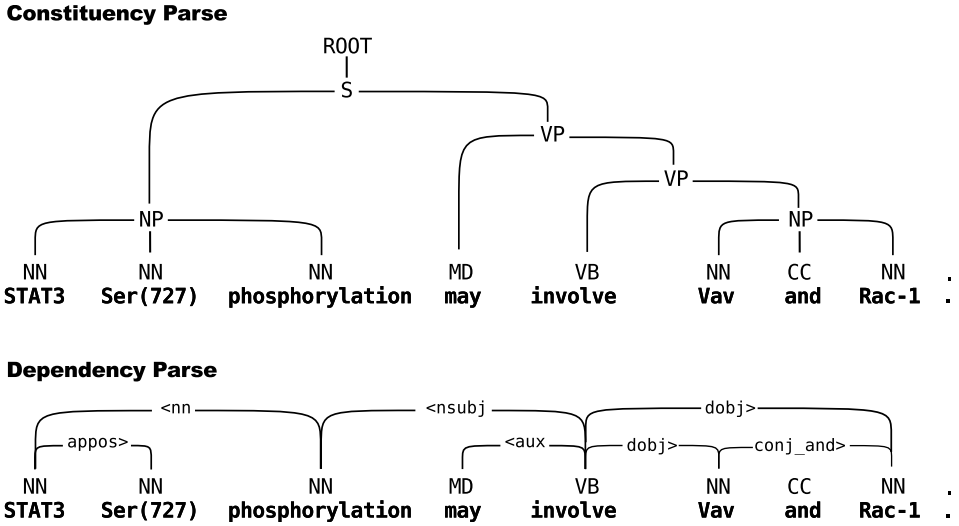


Figure 3.1: Constituency and dependency parses, shown for the same sentence. The constituency parse has intermediate phrase nodes whereas in the dependency parse each node corresponds to a single token.

Brown Corpus or the University of Pennsylvania (Penn) Treebank being popular in many systems. In modern parsers, POS tagging is part of the overall parsing process.

Where POS tagging defines only the leaves of the parse tree, a shallow parser aims to produce a limited higher structure, enough for many applications, but computationally less intensive than the generation of a full parse tree. Shallow parsing is also known as "chunking", referring to the process of taking the individual POS-tagged words, and forming larger phrases out of them. For example, "the new protein", a list of three tokens with the POS tags *determiner*, *adjective* and *noun* could form a single *noun phrase* when processed by a shallow parser. The Apache OpenNLP library contains a chunker that is sometimes used in biomedical text mining (Bui and Sloot, 2012).

When a shallow parse is not enough, a full parse tree can be generated for a sentence. In the parse tree individual tokens are the leaves, and other tokens or higher-level concepts such as phrases are their parent nodes. Such parse trees aim to fully define the relations of the tokens or phrases of a sentence. These relations are often syntactic concepts, but can also represent semantic roles. Two common parse tree schemes are the *constituency parse* and the *dependency parse*.

The *constituency parse* is based on a constituency grammar, characterized by multi-level nested phrases (See Figure 3.1). A constituency parse always has a root node. Branch nodes define the nested phrases of the

sentence, and the POS-labeled tokens are the leaf nodes. A constituency parse defines a very detailed breakdown of the elements of a sentence and can provide a lot of informative features for text mining, but producing such a parse is often a computationally intensive task. Most modern constituency parsers are statistical in nature, and can be re-trained for different domains and often even for different languages. Popular constituency parsers used in event extraction include the ENJU parser (Miyao et al., 2009), the BLLIP parser (Charniak, 2000; Charniak and Johnson, 2005) and the Stanford parser (Klein and Manning, 2003; de Marneffe et al., 2006).

For semantic information extraction, the constituency grammar can be somewhat difficult to utilize, as central subject-object relations are defined through multiple levels of nested phrases. The *dependency grammar* produces parse trees that are often much easier to use for tasks such as biomedical event extraction. A dependency grammar is built on the concept of the dependency relation, where the verb has a structurally central role, with all other words being dependent on it. In dependency grammars, phrase nodes do not exist, and all nodes are terminal (See Figure 3.1). These attributes cause a dependency parse tree to have much fewer nodes than a constituency parse, fewer levels of nesting, and more direct links for semantic relations such as *agent/patient*. The Stanford parser library provides a tool for converting a constituency parse tree into a dependency parse tree and this approach is also utilized in TEES. The Stanford system can produce several types of dependency parses. The *Stanford Dependencies (SD) collapsed form*, which further simplifies the parse, is the one used in TEES.

Being statistical in nature, modern parsers are also dependent on the parse tree corpora they have been trained on. Texts in different domains can differ quite a lot in terms of vocabulary and the types of phrases used, so a parser trained on e.g. newspaper text may not perform optimally on biomedical research text with its long, complicated sentences and specialized, domain-specific terminology. The machine learning aspect of parsers allows them to be adapted to new subdomains, and several efforts have been made to produce parsers more suitable for processing biomedical text. TEES uses the BLLIP parser with David McClosky's biomodel, a domain adapted model utilizing self-training and the GENIA tree corpus (McClosky, 2009).

3.2 The Graph Kernel

A graph kernel for PPI extraction was the first machine learning system developed at the University of Turku for biomedical text mining (Airola et al., 2008b). This project was the foundation for many of the methods used in TEES, and its technology continues to be utilized with good results by researchers working on PPI extraction (Thomas et al., 2011; Tikk et al., 2010).

The all-paths graph kernel was developed to automatically extract protein-protein interactions from the "five corpora" (AIMed, BioInfer, HPRD50, IEPA and LLL), commonly used PPI corpora converted to a unified binary relation format (Pyysalo et al., 2008). For each unordered protein pair in a sentence, the graph kernel attempts to predict whether it is an interacting pair.

The graph kernel implementation follows the theoretical graph kernel introduced by Gärtner et al. (2003). The graph kernel is calculated using an *adjacency matrix*. Each element corresponds to a directed path between two nodes in the graph. In the initial state of the matrix, an element can either be 1 (nodes are directly connected) or 0 (there is no direct connection between the nodes), meaning that the initial matrix contains all paths of length 1. When the adjacency matrix is multiplied by itself, the resulting matrix will contain paths of length 2, 3 and so forth, defined as the sum of the values of the component paths from earlier steps. An infinite sum of such matrices is calculated using the Neumann Series, resulting in the final adjacency matrix containing the summed weights of all possible paths connecting each pair of graph nodes. By modifying the initial weight of the direct connections between specific nodes, certain paths can be given more weight in the final adjacency matrix.

The PPI relations classified by the graph kernel are pairs of potentially interacting protein names (nodes) and the features used for classification are defined for each directly connected node pair, to be combined and weighted with the adjacency matrix.

The nodes of the graph used to construct the adjacency matrix are the word tokens in the sentence. These nodes are duplicated, as the graph consists of two subgraphs: The linear order of words in the sentence, where each word is connected to those before and after it, and the dependency parse, where words are connected via dependencies. In an approach following the shallow linguistic relation extraction method of Giuliano et al. (Giuliano et al., 2006), features generated from the linear subgraph are labeled as (B)efore, (M)iddle or (A)fter, relative to the candidate protein pair.

In an approach that would become a central feature in TEES, in the graph kernel dependency subgraph the *shortest path* is emphasized. That is, for each protein pair, the set of pairwise dependencies that provide the shortest path between the two proteins are given a higher initial weight in the adjacency matrix. Features on the shortest path are also labeled so that the classifier can make decisions based on the presence of specific syntactic elements on the shortest path.

In the graph kernel system, a variant of Dijkstra's algorithm (Dijkstra, 1959) was used to generate all shortest paths for a pair of nodes. In TEES, the Floyd-Warshall algorithm (Floyd, 1962), modified to likewise produce all shortest paths, is used. A filter is also added for skipping selected edge

types in order to produce paths more likely to capture interaction words. For example, in the DDI Extraction 2011 task the "conj_and" dependencies are skipped to avoid producing shortest paths with no intermediate nodes.

Other aspects of the graph kernel later adopted in TEES are very high dimensional, very sparse feature sets and the use of a high-performance machine learning classifier (RLS in the case of the graph kernel, SVM in TEES). Moreover, the use of Python for fast and flexible prototyping, and a highly parallel cluster system for optimizing the parameters of the classifier, are also approaches used in TEES.

3.3 The Shortest Path

The concept of the *shortest path* is a feature of the dependency parse utilized in relation extraction, based on the common assumption that words in the connecting nodes of the dependency path are especially relevant for the potential relation holding between the end nodes. E.g. Bunescu and Mooney define a formal *shortest path hypothesis* for this aspect of the dependency parse and build a *shortest path dependency kernel* focusing on information derived from these dependencies (Bunescu and Mooney, 2005).

In the context of event extraction, the shortest path hypothesis was analyzed in Paper I. Events, and the complex interactions of the BioInfer corpus, have many structural similarities with pairwise PPI relations. In particular, events with a specified *trigger* word can be defined as a set of pairwise relations. For example, the interaction "A binds B" might be defined as the (undirected) pairwise PPI relation *BIND(A, B)* between the two named entities. As an event, the same phrase would be defined as *BIND(trigger:binds, theme:A, theme:B)*, with the trigger word specified and argument roles labeled. However, considering the trigger word as a generic graph node comparable to the named entities, the same event could be defined as two pairwise relations, *BIND(trigger:binds, theme:A)* and *BIND(trigger:binds, theme:B)*, where the roles of the arguments could also be implicitly derived from the interaction type BIND, leading to two directed pairwise relations *BIND(binds, A)* and *BIND(binds, B)*.

This naturally leads to the TEES event extraction approach where events are divided into their component relations, enabling methods developed for PPI extraction to be directly applied to complex interactions. Further complications of course result from the more complex annotation, namely the need to detect not only named entities but also trigger nodes and to separate overlapping events, but largely this approach of implicitly "binarizing" event structures defines the core event extraction method used in TEES.

One feature that stands out when representing events as graphs is that events provide more intermediary nodes as compared to binary relations (See Figure 3.2). In the event representation, unlike a pairwise binary relation,

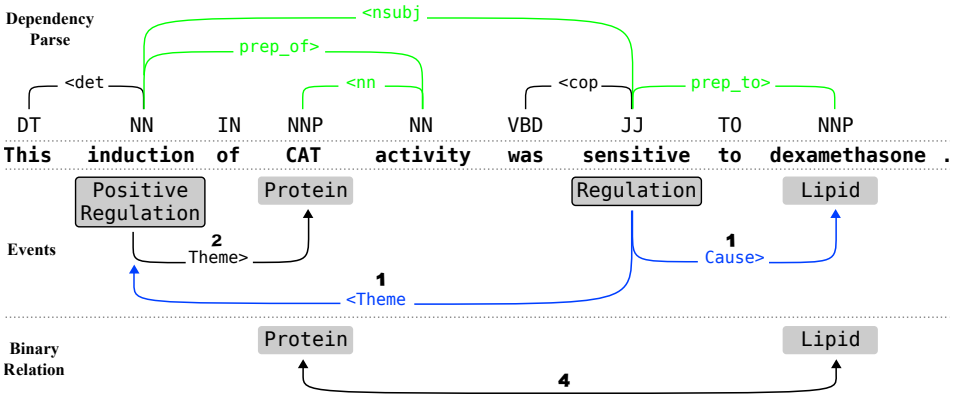


Figure 3.2: Event annotations share similarities with dependency parses. Event triggers form intermediary nodes on the *shortest path of dependencies* (shown in green), and event arguments often have a one-to-one correspondence with a dependency. In this example from the GENIA Event Corpus, there is a single corresponding dependency for both arguments of the *Regulation* event (shown in blue), while two dependencies exist between the endpoints of the *Positive Regulation* event's *Theme* argument. In contrast, four dependencies separate the endpoints, if the relationship is annotated with a binary relation (Figure adapted from Paper I).

the trigger word is included in the semantic annotation. Based on this observation, it was hypothesized that event arguments would correlate with the shortest dependency path more closely than binary relations, and this could aid in trying to extract them.

To compare event arguments with relations, a binarized version of the BioInfer corpus, where the complex nested interactions were interpreted into binary relations, was utilized (Heimonen et al., 2008). The BioInfer corpus consists of complex interactions which are largely similar in structure to events. However, they can also describe physical relations (such as substructures) unlike events, which are usually considered to describe active processes. The binarization of the BioInfer corpus converts these event-like structures into pairwise binary relations, resolving nested structures with a set of rules. When compared to the arguments of complex interactions, the shortest path for binary relations consists on average of considerably more dependencies, demonstrating that event-type annotations correlate more closely with a dependency parse (See Figure 3.3). This correlation is noticeably strong with dependency parses of the collapsed Stanford format, which has been developed with semantic mining applications in mind. Using this parse, over 60% of GENIA and BioInfer event or complex interaction arguments have a single, corresponding dependency.

As with binary relations, event arguments link together words that can

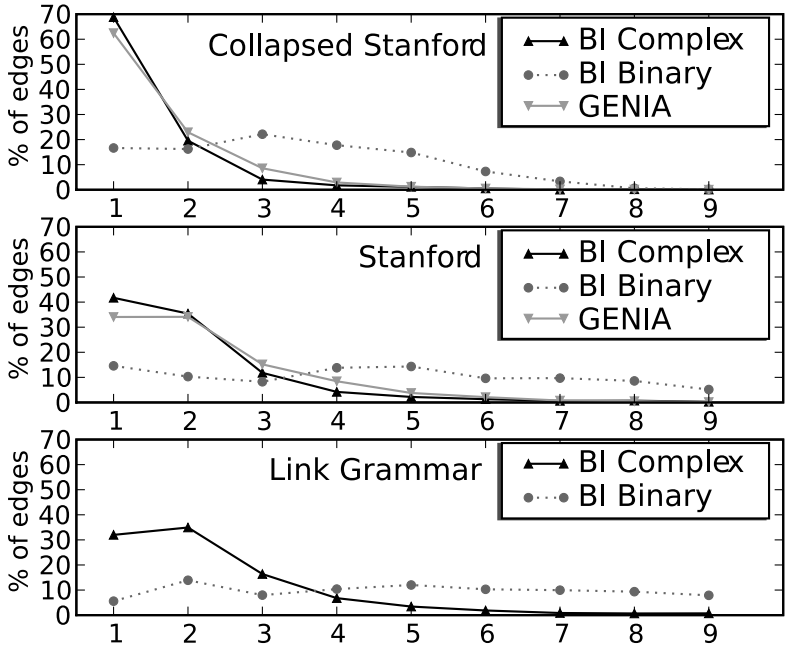


Figure 3.3: GENIA corpus event arguments and comparable BioInfer corpus complex interactions are more likely to correspond to a single syntactic dependency than pairwise relations from the binarized BioInfer corpus (Figure from Paper I).

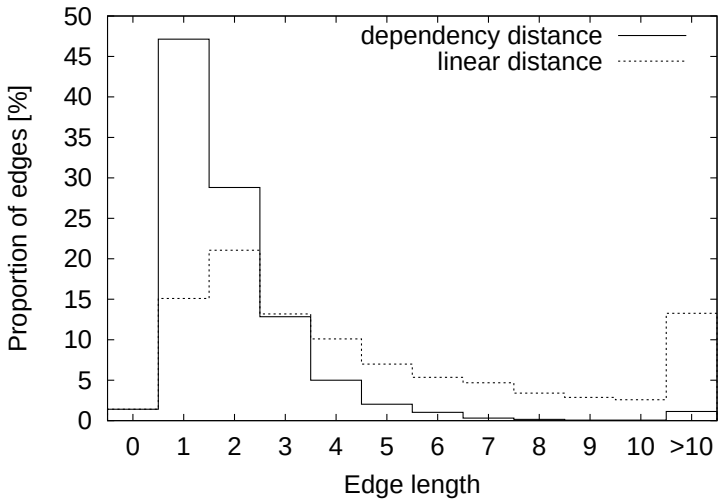


Figure 3.4: The number of dependencies on the shortest path compared with the number of words separating the two terminal words of an event argument (Figure from Paper II).

be distant in the linear order. As the dependency parse, especially the collapsed Stanford variety was shown to correlate strongly with the event arguments, the BioNLP'09 Shared Task GENIA corpus was analyzed to determine how close event arguments are to the dependency parse. As can be seen from Figure 3.4, almost 50% of all GENIA event arguments correspond to a single collapsed Stanford dependency, and the vast majority take a maximum of three links. Compared to the linear order, where only 15% of event arguments are at a distance of one word, and more importantly, where more than 10% of event arguments travel over more than 10 words, the dependency parse is very similar to the semantic annotation of the events.

3.4 Entity Syntactic Heads

TEES utilizes the dependency graph to predict the event annotation graph. To use the correlation, the two graphs must be aligned, so that one can say which edge or path of edges corresponds to which event argument. In one graph, the nodes are tokens and edges are dependencies, in the other, the nodes are named entities or triggers and edges are event arguments. To connect these graphs, their nodes have to be aligned. The dependency graph is always known information, so the two graphs are connected by mapping the named entity or trigger nodes of the event graph to the token nodes of the dependency graph.

In many cases this mapping is trivial, for example a trigger node *Phosphorylation* with the text "phosphorylates" can easily be mapped to the single token "phosphorylates". However, in corpora like GENIA named entities and triggers can span multiple tokens, for example the "human IL-4 protein" spans three separate tokens. In such cases, the whole named entity or trigger is mapped to the *syntactic head token* of the parse subtree it covers. The head token is determined by giving each token a *head score*, then picking from the tokens corresponding to the trigger or named entity the one with the highest score. If two or more tokens share the same highest score, the rightmost one in the linear token order is used.

The head score algorithm is specific for the Stanford scheme dependency parse. All tokens are given an initial score of 0, except for tokens "\", "/", or "-" which get a score of -1, as they are likely a byproduct of splitting named entity tokens, and in such cases a more informative substring should be the head. After initial scores are defined, the algorithm loops over all dependencies of types commonly encountered in multi-token entities[2]. If the governor token of such a dependency has a score lower or equal to the score of the dependent token, the score of the governor token is increased by one. All dependencies are reprocessed as long as any score is modified or until the maximum number of iterations (20) is reached, indicating a potential loop. This algorithm is designed to survive the loops and broken parses often produced for complicated sentences, and to produce a reasonable ordering for the tokens with the information available.

Occasionally a single token can include several named entities or triggers. We use a rule-based "protein name splitter" that divides these tokens at punctuation. For example, the token "p50/p65" contains two named entites, and the token "GATA3-binding" contains a named entity and a trigger. Without this preprocessing step, multiple entities or triggers would end up having the same syntactic head token, preventing detection of events between them.

3.5 Features for Event Extraction

The analyses on dependency parse correlation with event annotations were the basis for the approach used in TEES to derive features from the parse for machine learning. The shortest path is the primary source of features for interactions, whether they are binary relations or event arguments. For triggers and other nodes, the immediate dependency context is used to define features. TEES differs from the graph kernel in that the whole dependency graph is not used for features, but rather specific regions are utilized to produce task-specific features. Compared to the graph kernel, this makes

[2] *prep, nn, det, hyphen, num, amod, nmod, appos, measure, dep* and *partmod*

TEES as a system more specialized for text mining, reduces the number of features, and therefore makes processing less computationally intensive. The downside of this approach is that instead of a single technique based on a well-defined mathematical theory, the syntax representation for machine learning is constructed from a series of solutions arrived at by trial and error. However, in the context of current relation and event extraction tasks, the TEES approach has been shown to perform well and remains generalized enough to be easily adapted to different extraction targets.

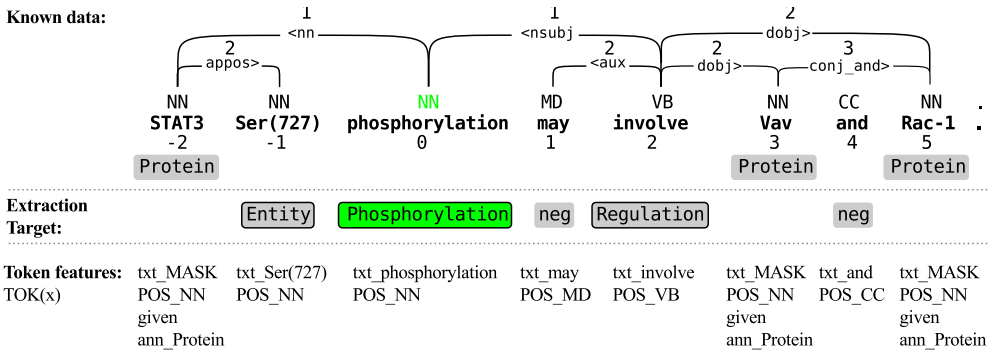
TEES consists of different modules for the different classification steps in event extraction, such as trigger detection, edge detection or modifier detection. These modules use different features, but most of them are derived by combining a set of simple, core features in a variety of ways. The primary feature groups that form the building blocks for more complex features are *dependency features* and *token features*. The dependency features are the *type* of the dependency (such as nsubj or aux) and its *direction* relative to its two token nodes or a longer path of dependencies. The basic token features are the *text* of the token, its *POS* type and the *types of the known entities* which it is part of. The token features can be extended with task-specific features, such as the text being a known speculation word, and these new component features will then produce further variants of the combinatoric features built from them.

We will next analyze the four classification steps of TEES event extraction and see how their feature representations are used to describe the syntax and other known data for the classifier.

3.6 Entity Detection

Entity detection refers to the process of marking the boundaries of named entities and event triggers in a given sentence to produce the nodes of the event graph. This part of TEES is conceptually similar to the NER (Named Entity Recognition) field of research, but is simplified with the limitation that each predicted entity consists of a single token. In training data, where annotated entities can have arbitrary (even disjoint) spans, each entity is first mapped to a single token, as explained in Section 3.4.

Trigger detection becomes thus a classification task relating to a single word token and the chosen feature representation reflects this. In classifying the token, the token itself is of course of central importance, and TEES attempts to derive as much information as possible from it. As shown in Figure 3.5, a large number of features are generated from the token. The basic *token features* generated for each token are of course used also for entity detection. The Porter Stemmer (Porter, 1980) is used to derive the stem of the word token. The stem and the non-stem part of the word are used



Example #2 (positive of class "Phosphorylation") features:

Sentence Features

Given entity count: given_count_3

Sentence bag-of-words with counts: STAT3=1, given_STAT3=1, Ser(727)=1, phosphorylation=1, may=1, involve=1, Vav=1, given_Vav=1, and=1, Rac-1=1, given_Rac-1=1

Root Token

Token features: TOK(phosphorylation)

Porter Stemmer features: stem_phosphoryl, tail_ation

Normalized text (remove -, /, ,, \, and whitespace) features: nor_phosphorylation, norstem_phosphoryl, nortail_ation

Substring (split at '-') features: sub_phosphorylation, substem_phosphorylat

Content features: upper_case_start=0, upper_case_middle=0, has_digits=0, has_hyphenated_digit=0, has_hyphen=0, has_slash=0, has_backslash=0

Duplets and triplets: dt_ph, dt_ho, dt_os, dt_sp, ... tt_pho, tt_hos, tt_osp, tt_sph, ...

Linear Context

Linear order features in range [-3,-1] and [1,3]: TOK(STAT3)_L-2, TOK(Entity)_L-1, TOK(may)_L1, TOK(involve)_L2, TOK(Vav)_L3

Linear *N*-grams in range [-2, 0]: MASK_ser(727)_phosphorylation, ser(727)_phosphorylation

Dependency Context

Dependency context: nn_1, nsubj_1, appos_2, aux_2, dobj_2, conj_and_3

Dependency context: TOK(Stat3)_d1, TOK(Ser(727))_d2, TOK(may):d2, TOK(involve):d1, TOK(Vav)_d2, TOK(Rac-1)_d2

Dependency chains: d1_nn>, d1_<nsubj, d2_nn>_appos>, d2_<nsubj_aux>, d2_<nsubj_dobj>, d3_<nsubj_dobj>_conj_and>, d3_<nsubj_dobj>_<conj_and

Dependency chain given entities: d1_nn>_given, d2_<nsubj_dobj>_given, d3_<nsubj_dobj>_conj_and>_given, d3_<nsubj_dobj>_<conj_and_given

Figure 3.5: Entity Example Builder Features. One example is generated for each syntactic token that does not belong to a *Protein* entity. This sentence produces three positive and two negative examples. The numbers show the distance of dependencies and tokens from the highlighted example.

as features that can detect the same word in different inflected forms. The text is also normalized, to combine cases like "coimmunoprecipitate" and "Co-immunoprecipitate". The content of the word is analyzed for features that might be typical for biomedical keywords, such as upper casing or presence of hyphens and digits. The potential head token is also split into two- and three-letter duplets and triplets for a detailed representation of its structure. All of these features that aim to either generalize or subdivide the basic features of the token never replace the basic features, but are generated in addition to them. Therefore, for a token "phosphorylation" both the full word "phosphorylation" as well as the stem "phosphoryl" are represented in the features, leaving the choice of how specific features are used for learning up to the classifier.

As with the other components, the trigger detector produces for each example also a representation of the full sentence it exists in, using an unordered bag-of-words and the number of given entities (usually protein names from a separate NER step) already detected for the sentence.

In using the syntactic parse, the trigger detection system shares a lot of similarities with the edge detection one, not least because the edge detection system was developed first and the work on trigger detection naturally utilized methods already shown to work. The concept of the shortest path does naturally not apply to a single token, so instead the trigger detector builds a representation of the *dependency context* of the token, following all undirected dependency paths up to a depth of three leaving from this token. This process generates a large number of features and aims to capture such relations as a trigger token being the governor of a *dobj* type dependency connecting to the head of an already detected protein.

As shown in Björne et al. (2011b) entity detection cannot be performed simply with a dictionary lookup, as many trigger words are sometimes trigger entities, sometimes not. Furthermore, the same word can in one sentence be a trigger of one type, in another a trigger of another type. For example, in the 2009 BioNLP Shared Task GENIA corpus the token "overexpression" is roughly evenly distributed among *Gene expression*, *Positive regulation* and the negative class.

In TEES version 2.1 the trigger detection step generates for the BioNLP'11 Shared Task GENIA corpus training and development sets around 200,000 examples with a feature space of around 540,000 features, with around 110 features per example on average. All TEES classification steps have a very sparse feature space, as is common in text mining. With modern implementations of linear SVMs this amount of data remains easily processable.

The main limitation of the TEES approach to entity detection is the heavy dependence on features derived from the syntactic parse. While the parse is a rich source of features, parsing a sentence with the best parsers currently available for the biomedical domain is a very time-consuming pro-

cess, and in applications such as PubMed-scale text mining producing a deep parse for each sentence can become computationally unfeasible. In most event extraction tasks TEES relies on a preprocessing step where a dedicated, fast NER-system, such as BANNER, is used to detect gene or protein names (which form the leaf nodes of the event graph), therefore limiting the parsing (and subsequent event extraction) to sentences with at least one such entity detected. While TEES has been adapted to extraction targets where its trigger detector extracts all required entities, such as the BioNLP'11 Bacteria Gene Interactions tasks or drug–drug interaction extraction, the requirement to parse each sentence limits the usability of such models on large-scale datasets.

3.6.1 Variations on Entity Detection

The basic structure of the entity detection system has remained the same since the original TEES version from 2009, although the feature representation has been somewhat optimized. Moreover, for the BioNLP'11 Shared Task several task-specific modifications were developed to address the particular requirements of different domain targets.

For the BioNLP'11 Shared Task, external databases were for the first time used as an additional source of features. Such real-world contextual knowledge holds great promise in improving text mining, but at the risk of making systems overly reliant on what is already known, thus making it harder to detect examples not already in the databases. Therefore, the use of external data in information extraction projects should always be carefully considered. In the BioNLP'11 and BioNLP'13 Bacteria Biotopes tasks the TEES entity detector was enhanced with existing knowledge from the List of Prokaryotic names with Standing in Nomenclature (LPSN) (Euzéby, 1997) and Wordnet (Fellbaum, 1998).

To specialize the TEES entity detector for different tasks, the easiest way to introduce additional features is to add them to the common *token features* (See Figure 3.5) from where they will be combined into many of the more complex features. The LPSN database maintains a list of names for prokaryotes and was used to help in detecting *Bacterium*-type entities by extending the *token features* with a binary feature marking the presence of the token in the set of known bacteria name tokens. Synonyms and hypernyms from Wordnet were used in an attempt to detect similarities between the very heterogeneous *Environment*-entities describing elements where bacteria live. For example, both "chicken" and "milk" are *Food*-entities, and while their character strings have nothing in common, both have the same WordNet hypernym "noun.food".

In the case of the BioNLP'11 EPI (Epigenetics) task, types of reversed trigger entities were merged with their direct forms. For example, *phospho-*

rylation and *dephosphorylation* were merged into a single *phosphorylation* class. After entity detection, a rule-based step was used to separate the reverse forms. This step was shown to determine a reverse class correctly for 99.6% of cases in the EPI training dataset. As many of the reverse classes were quite small and thus difficult for the SVM to learn, this approach increased performance on the EPI task by 1.33 percentage points and made it possible to extract several of the small reverse classes. For example, the previously undetectable *deubiquitination* class with 8 instances in the development set was predicted with an F-score of 77.78%.

In two cases the trigger detector has been modified to detect entities consisting of more than a single token. In the BioNLP'11 Coreference supporting task the detected entities had to cover at least a minimum span, a part of the entity that could consist of multiple tokens. To produce suitable entity candidates, the syntactic phrases from the BLLIP-parse, further subdivided with a set of rules, were used as entity candidates. This resulted in a very large number of examples, mostly negatives. In the BioNLP'11 Bacteria Biotopes task exact spans were required for *Bacterium*-type entities, so a simple rule-based system was developed to expand the detected single-token entities forwards and backwards from the head token largely based on a dictionary of known bacterium name tokens extracted from LPSN. This rule-based step was shown to have 91% accuracy on the training data. Both of these multi-token entity detection approaches were quite specific for the tasks they were developed for and have not been used in other work.

3.7 Edge Detection

The edge detector is the first TEES component developed (Björne et al., 2009a). In this work, it was evaluated for extraction of event arguments from the BioInfer and GENIA corpora and shown to have higher performance on this task than the graph kernel, the system developed earlier for PPI extraction. The classifiable example produced during edge detection is most often a directed, typed edge linking together two entity nodes. Since entities are linked to the parse through their syntactic head tokens, edges can have a corresponding path of dependencies (See Figure 3.6).

The edge detector relies largely on features generated from the shortest path of dependencies. This shortest path is subdivided into shorter sections called N-grams, to produce features more likely to be present in multiple sentences (See Figure 3.6). It is important to note that the edge detector uses features generated from entities produced by the entity detector, linking its performance to this earlier event extraction step. Given gold standard entities, edge detection has often significantly higher performance, indicating that the system relies heavily on the correctness of the detected entities (Björne et al., 2011b).

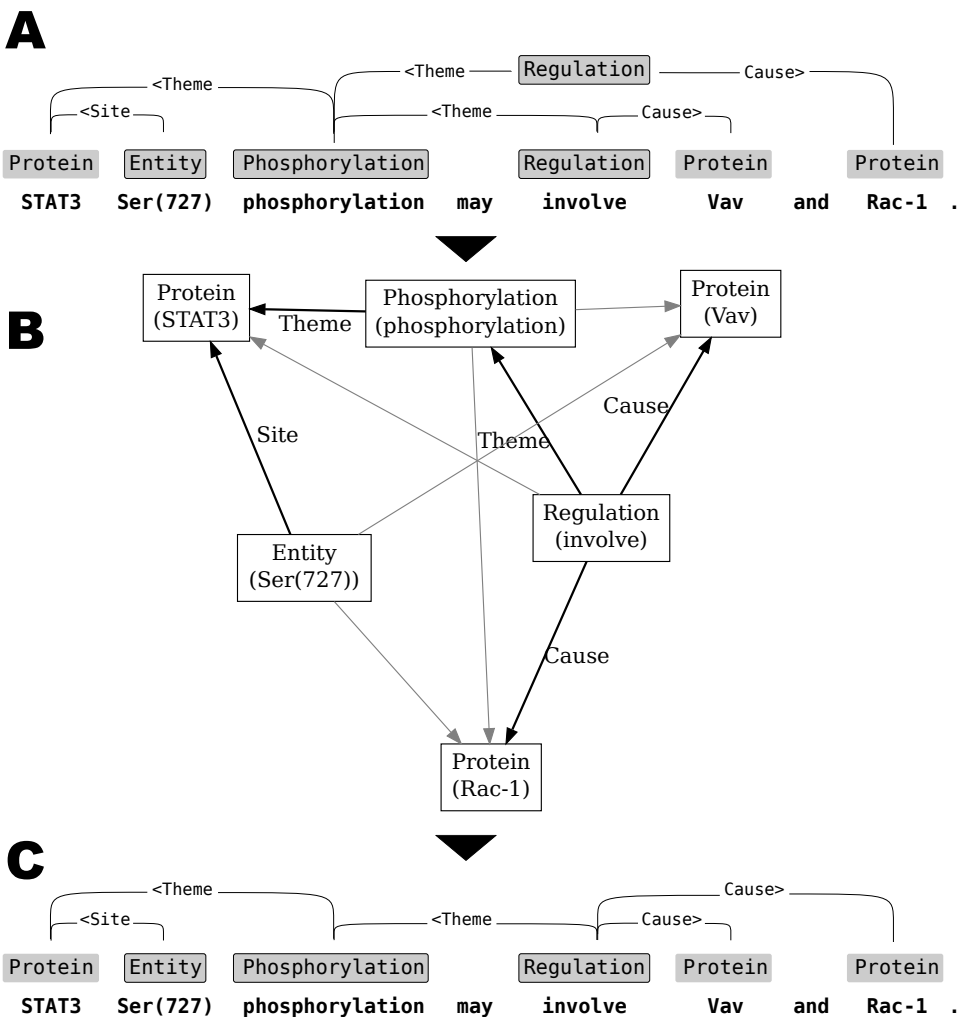
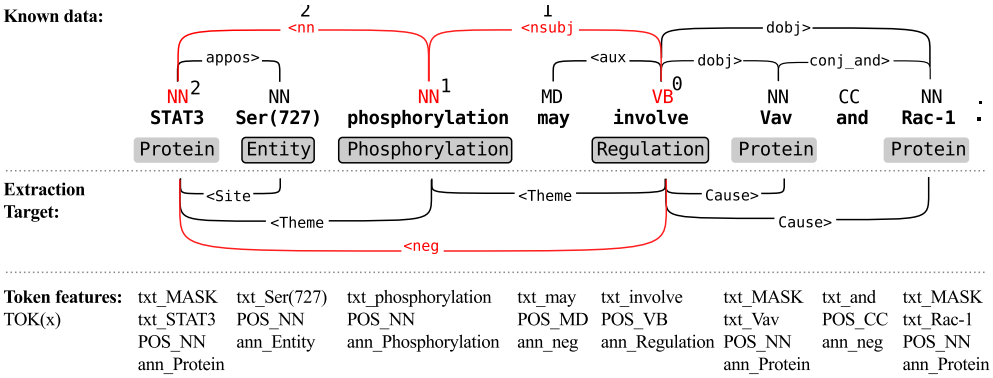


Figure 3.6: Edge Detection. The Edge Example Builder builds one classifiable example for each potential edge. A gold standard annotation (A) can be used to label the examples. Potential edges exist between all tokens where an event argument or relation could connect the entities the heads of which the tokens are (B). As only a single trigger entity can be produced by the entity detector for each token, overlapping events are predicted in a merged format (C).



Example *Regulation(involve)→Protein(STAT3)* (negative example) features:

Sentence Features

Entity counts: n_Protein=3, n_Entity=1, n_Phosphorylation=1, n_Regulation=1

Entity Features

Features ENT(X) generated by Entity Example Builder (See Figure X): e1_ENT(involve), e2_ENT(STAT3)

Entity Pair Features

Entity features: e1_TOK(involve), e2_TOK(STAT3), e1_predicted_e2_given, eTypes_Regulation_Protein
Terminal token features: t1_TOK(involve), t2_TOK(STAT3)

Shortest Path Feeatures

Single element features: nsubj>, nn>, internal_txt_phosphorylation, internal_POS_NN
Annotated type features: path_has_Regulation, path_has_Phosphorylation, path_has_Protein
Elements: dep_nsubj, dep_nn, tok_TOK(involve), tok_TOK(phosphorylation), tok_TOK(STAT3)

Shortest Path N-gram Features

Edge directions *N*-gram: directions_>>
Entity type *N*-grams in range [0, n-1]: Regulation_Phosphorylation
***N*-gram component roles:** directions_>>_TOK(Phosphorylation), directions_>>_pos0_nsubj, directions_>>_pos1_nn
Dependency *N*-grams in range [0, 4]: directions_>>_nsubj_nn
Dependency *N*-grams with terminus entities: Regulation_directions_>>_nsubj_nn_Protein
Governor/Dependent 2-grams: gov_TOK(involve)_dep_TOK(phosphorylation), gov_TOK(phosphorylation)_dep_TOK(STAT3)
Entity/dependency/entity 3-grams: ann_Regulation_nsubj_ann_Phosphorylation, ann_Phosphorylation_nn_ann_Protein

Figure 3.7: Edge Example Builder Features. This sentence produces five positive and five negative examples, as shown in Figure 3.6. The numbers indicate the distance of dependencies and tokens along the shortest path of dependencies corresponding to the highlighted example.

3.7.1 External Knowledge for Edge Detection

Similar to the entity detector in the BioNLP'11 Bacteria Biotopes task, the edge detector was enhanced with external database knowledge in the Bacteria Gene Renaming (REN) task. The task consisted of detecting pairwise relations marking statements where a new *B. Subtilis* gene name was introduced to supercede an old one. Most of this information was already covered in external databases, so pre-existing pairs were extracted from the UniProt *B. Subtilis* gene list "bacsu"[3] and *Subti*Wiki, the *B. Subtilis* community annotation wiki[4]. It was shown that for the 300 gene renaming relations in the training data, the UniProt list contained the pair for 66%, *Subti*Wiki for 79% and a union of both sources for 81% of the relations. Conversely, only 2.1% of negative examples had a corresponding pair in the union set.

Unsurprisingly, adding the presence of a known, annotated pair as a feature increased performance from 67.85% to 87.0%, leading to the best performance in the task by a margin of 17.1 percentage points. While this example shows that external knowledge can give a huge boost for a text mining system, it is also a good example of the dangers of using such knowledge. With the external knowledge, the system relies heavily on the list of known pairs, likely making it less capable of detecting yet unknown renaming statements. The question with using external data then becomes whether one wants to find all textual mentions of a set of known facts, or whether one wants to detect completely new events of a given type. In the first case, external knowledge is a clear benefit, in the latter a potential hindrance, but of course the balance to be found depends on each particular task.

The REN task was a case of pairwise relation extraction, presented as a supporting task of the event-oriented BioNLP Shared Task. TEES has however been evaluated also on a pure relation extraction task, the 2011 First Challenge Task on Drug-drug interaction extraction (DDIExtraction11), as described in Paper V. The goal of the task is to detect statements on adverse effects between given drug mentions in text. These relations are untyped (they have a single type) and undirected, making the task structurally similar to traditional PPI relation extraction tasks. In this task, only the TEES edge detector component was needed, and it was optimized for this particular challenge. Apart from task-specific features, the use of binary classification allowed the application of "thresholding", modifying the balance between positive and negative classes to increase performance. The Turku Event Extraction System placed fourth, four percentage points behind the leading system by team WBI of Humboldt-Universität Berlin (Thomas et al., 2011; Segura-Bedmar et al., 2011).

[3] `www.uniprot.org/docs/bacsu`

[4] `subtiwiki.uni-goettingen.de/wiki/`

Most of the specialization towards the DDIExtraction11 dataset was done by adding features from external datasets. The task organizers provided MetaMap analyses for the task corpus, produced with the UMLS MetaMap Transfer (MMTx) tool (Aronson, 2001). From the MetaMap analyses, additional features given to drug entity head tokens included predicted long and short names, prediction probabilities, semantic types and CUI numbers. Binary features were added to mark if an annotated drug mention had not been detected as a known drug by MetaMap, and whether the two drugs in the candidate pair referred to the same MetaMap instance.

Additional features were built from DrugBank, the database which the DDI11 corpus is based on. Unlike the corpus, which marks instances of interaction statements in text, the DrugBank database simply lists known interaction pairs for each drug. For each candidate pair additional features were added to mark whether both drug names are present in DrugBank and whether DrugBank lists them as a known interaction pair.

Adding the DrugBank features increased performance by 0.94 percentage points, and adding also MetaMap features gave a further increase of 0.99 percentage points. As with the Bacteria Biotopes task, external data was shown to improve text mining performance, but this again comes at the risk of making the system good at detecting only the known cases. The MetaMap features, only adding more information about the given drug entities, probably do not have much risk of this. The known interaction pairs from DrugBank, even if they can't tell whether each specific mention of such a pair in text is an interaction, are already a greater risk when using the system in real-world applications. As with the Bacteria Biotopes example, the tradeoff is likely to be high performance for detecting instances of known interactions vs. lower overall performance, but a higher chance of detecting new interaction pairs.

3.7.2 Modifying the Shortest Path

The TEES edge detector was developed for detecting event arguments, which connect a trigger word to another trigger word or named entity. In such edges, the most important words are at the ends of the shortest path. However, in pairwise PPI relation extraction where the trigger word is not explicitly marked it may not fall on the shortest path between the two named entities, even if it would provide essential information about the interaction. In the DDIExtraction11 binary relation extraction task, to avoid situations where the shortest path would become trivially short, *conj_and* dependencies were excluded from the path construction (See Figure 3.8). This simple customization increased F-score on the development set by 0.42 percentage points.

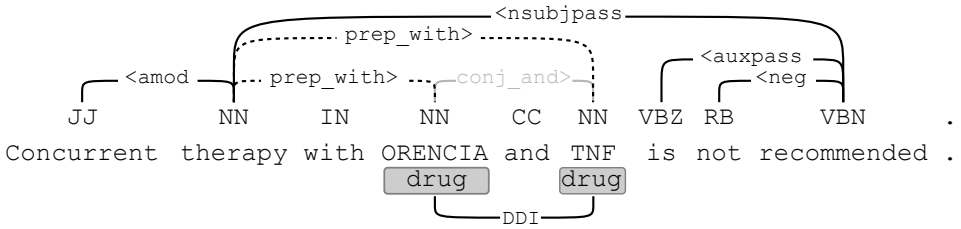


Figure 3.8: By ignoring *conj_and* dependencies when constructing the shortest path, words essential for describing the interaction can more often be included in the path. Here the word "therapy" fulfills a role similar to event corpus triggers. Figure from Björne et al. (2011a).

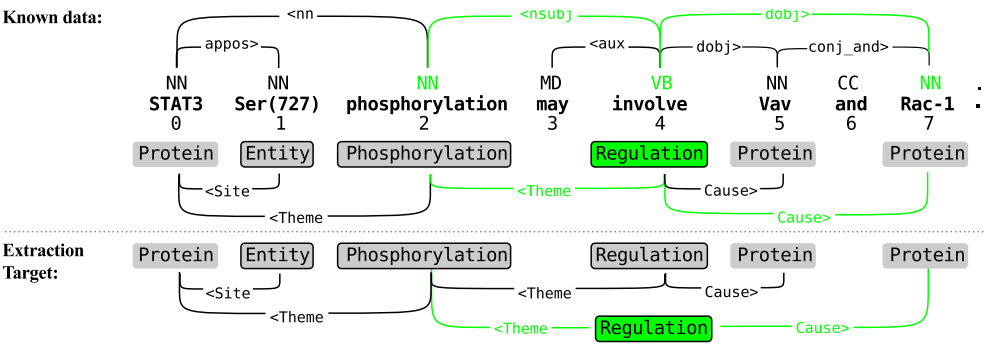
This modification is of course specific for the parse, and due to time constraints, edge filtering beyond *conj_and* was not studied much, but based on this result it can be said that by artifically extending the shortest path in this manner, an approach originally developed for event argument detection can be better optimized for pairwise relation extraction tasks. In TEES version 2.1, dependency types to be filtered can be defined in the command line parameters passed to the edge detector.

3.8 Unmerging

The unmerging detector is the component that takes the merged event graph produced by the entity and edge detectors and duplicates event nodes in order to separate arguments into valid combinations. The resulting event graph can be directly interpreted as BioNLP Shared Task -like events.

The unmerging step was originally implemented for the BioNLP'09 Shared Task as a rule-based system. The system removed invalid edges, broke argument loops and used a set of heuristics based on the dependency parse to group event arguments (Björne et al., 2011b). The rules of this system were specific for the BioNLP'09 GENIA corpus, so in further research a decision-tree-based machine learning system was tested for use on both the GENIA and BioInfer corpora (Heimonen et al., 2010). In the 2011 BioNLP Shared Task and TEES versions since then, unmerging has been implemented as a classification task similar to other processing steps.

The system looks at each event trigger node, and produces one example for all valid outgoing edge combinations (See Figure 3.9). These are classified as positives or negatives, optionally with multiclass classification for tracking event type specific performance. After classification, the unmerged events are combined again into a connected graph, further duplicating nested nodes as required.



Example *Regulation(Theme:Phosphorylation(phosphorylation), Cause:Protein(Rac-1))* (positive example) features:

Linear Span Features

Bag-of-words between arguments: bow_range_5, bow_range=5, nonentity_txt_may, nonentity_txt_and, slash_or_hyphen_in_bow=0 *(bow features get a special tag if bow range is 1, i.e. only one word separates the arguments)*

Argument Combination Features

Argument role features: arg_Theme, target_Phosphorylation, target_nongiven, pair_Theme_Phosphorylation, arg_Cause, target_Protein, target_given, pair_Cause_Protein, context_Cause, context_target_Protein, context_target_given, context_pair_Cause_Protein

Count features: arg_count_2, arg_count=2, all_arg_count_3, all_arg_count=3, arg_Theme_count_1, arg_Theme_count=1, arg_Cause_count_1, arg_Cause_count=1, con_Cause_count_1, con_Cause_count=1

Argument Content Features

Entity features ENT(X) generated by Entity Example Builder (See Figure X): trigger_ENT(involve), Theme_ENT(phosphorylation), Cause_ENT(Rac-1), context_Cause_ENT(Vav)

Argument edge features EDG(X) generated by Edge Example Builder (See Figure X): Cause_EDG(involve, Rac-1), Theme_EDG(involve, phosphorylation), context_Cause_EDG(involve, Vav)

Figure 3.9: Unmerging Detection Features. The merged event graph shown in the figure produces five examples. The three positive examples are for *Phosphorylation(Theme:STAT3)*, *Regulation(Cause:Vav, Theme:Phosphorylation)* and *Regulation(Cause:Rac-1, Theme:Phosphorylation)*. The two examples *Regulation(Theme:Vav)* and *Regulation(Theme:Rac-1)* are negative, as even if they are structurally valid Regulation events (Regulations can be without a Cause-argument) they do not correspond to the annotated gold events. The numbers in known data show the token indices, where tokens 3–6 form the argument span bag-of-words.

In TEES 2.0 the valid argument combinations defining the candidate events are encoded in the unmerging example builder and are limited to detecting event structures defined in the eight tasks of the BioNLP'11 Shared Task. Starting from TEES 2.1, valid argument combinations are automatically learned from the training data, enabling the system to be used on new annotation schemes with no additional programming required.

A candidate event example consists of a trigger node and outgoing edges, which link it to other trigger or named entity nodes. To describe this structure as features, the system relies on the entity and edge prediction feature representations (See Figure 3.9). A feature representation similar to the one used in entity detection is used to describe the trigger node and the other nodes linked by the outgoing edges, both the ones that are part of the candidate event and the context edges that are not. The edge detector feature representation is used to describe the outgoing edges. To distinguish the features of the different parts of the event, entity and edge features are labeled with the type of the edge or entity and with a further label if they are not part of the current event candidate.

Features specific for the unmerging detector include a bag-of-words covering the span of text between the most distant argument targets of the event, in an approach reminiscent of the jSRE relation extraction system (Giuliano et al., 2006). Additional features are defined to mark the presence of entity and edge types both in the candidate event and other outgoing edges of the trigger node. Edge and entity type are also merged to define the presence of specific combinations. Finally, argument count features denote the number of different types of arguments.

An issue for training the classifier for unmerging detection is that while examples can be generated from a merged and unmerged gold annotation corpus, in actual predicted data there are also false positive edges and nodes. To train the system on more realistic data, a second copy of the training data is used, for which edges and entities are predicted using models trained on the same dataset. Classifying the training data with models trained on it results naturally in very high performance, so only a small number of incorrect entities and nodes is included in the extended unmerging training data. Even so, unmerging performance for predicted datasets is slightly improved. The additional unmerging examples could of course be generated by e.g. using the development set for training separate edge and entity detector models and using these to re-classify the training set, but this would require two more rounds of parameter optimization and make the already complicated multi-stage system even slower.

Apart from task-specific event candidate limitations, the unmerging detector has not been specialized for the different tasks. This is mostly due to the fact that all the tasks where external data has been used have been pairwise relation extraction tasks which do not require the unmerging step.

3.9 Modifier Detection

The final component in the TEES stepwise event extraction process is modifier detection. Modifiers define additional attributes for events, such as the *negation* and *speculation* modifiers used to mark event modality in some BioNLP Shared Task corpora. Modifier detection falls outside the graph prediction task of the main TEES event extraction approach, and is performed as an additional post-processing step on the final, unmerged event graph.

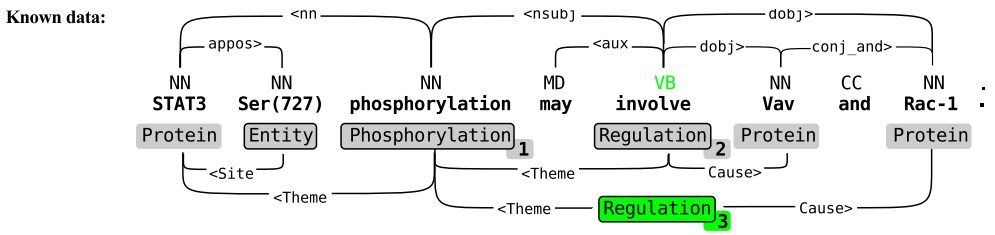
Each event trigger node generates an example to be classified for the presence of modifiers. Overlapping modifiers are rare, and with only two modifier types, merged classes can be used in these cases. It is also possible to perform modifier detection as multiple binary classifications, generating one example per trigger node per modifier type.

The TEES modifier detector uses a modified version of the entity detector, similarly building a feature representation based on the event trigger node. Sentence level features include node counts and a bag-of-words (See Figure 3.10).

The most important feature specific for modifier detection is a manually curated list of 115 speculation-specific key words such as "aim", "findings" and "shown". This list was compiled in 2009 from the BioNLP'09 Shared Task GENIA corpus and was first used in Paper II. The speculation words are used as token features, leading to their incorporation into the more complex features such as the dependency context features.

The TEES negation and speculation detector used in the BioNLP'11 Shared Task was jointly trained for the GE, EPI, and ID tasks. This simplified processing, important in the limited time available for system development, but resulted in potentially sub-optimal performance. In later experiments the modifier detector was re-trained using only the GENIA data, resulting in a gain of 2.42 percentage points from the previous F-score of 26.86% (which was already the best in the 2011 shared task). This approach of training modifier detection specifically for each task is used in TEES version 2.1.

Modifier detection has always been left in TEES as something of an afterthought. Partially this is due to the way modifier detection has been evaluated in the BioNLP Shared Tasks, as an additional attribute of predicted events, meaning that modifier detection performance is largely dependent on how good the system is at detecting events and as such efforts have been better spent in improving general event extraction performance. Compared to event detection, modifier detection may feel like a minor detail, but modality such as *negation* can have a large impact on the meaning of extracted event data.



Example #3 (positive of class "speculation") features:

Sentence Features

Entity counts: given_count_3, predicted_count_4
Sentence bag-of-words with counts: STAT3=1, given_STAT3=1, Ser(727)=1, pred_Ser(727)=1, phosphorylation=1, pred_phosphorylation, may=1, known_speculation_word_may=1, sentence_has_speculation=1, involve=1, pred_involve=1, Vav=1, given_Vav=1, and=1, Rac-1=1, given_Rac-1=1

Entity Features

Features ENT(X) generated by a simplified version of the Entity Example Builder (See Figure X): trigger_ENT(involve)

Figure 3.10: Modifier Detection Features. Events can have modifiers, such as *speculation* or *negation*. Modifiers are predicted for each event trigger node, either as a binary classification for each modifier type, or as shown here, with multiclass classification where overlapping modifiers are represented with merged classes such as *negation/speculation*.

The original GENIA modifier annotation scheme has later been extended into a general meta-knowledge annotation, and systems have been developed for extracting this information (Thompson et al., 2011; Miwa et al., 2012) As a growing part of event extraction, retrieving such meta-knowledge can be an interesting future area of research. If the TEES implementation were to be further improved, likely directions would be adding a list of keywords for *negation* and other modifiers, and including the event arguments in feature generation.

Chapter 4

PubMed-scale Event Extraction and Applications

The Turku Event Extraction System was developed for the BioNLP Shared Tasks, where participants could focus on just the event extraction, with named entities, syntactic parses etc. provided by the organizers (Kim et al., 2009; Stenetorp et al., 2011). With the good performance reached in the shared tasks, applying TEES to real-world text mining tasks became relevant, as shown in Paper III. To make such work possible, additional NLP tools were required to produce the supporting data given in the shared tasks, and also a corpus for mining biomedical events from was needed.

The corpus part was easily solved: the PubMed database is provided in downloadable format, so the entire set of (in 2010) 17.8 million citations was chosen as the target. No domain limitations were defined to ensure a realistic dataset, even if the inclusion of citations falling outside the biomolecular scope (such as medical patient cases) would result in some additional false positives. With the wide coverage of PubMed, the dataset was assumed to provide extensive coverage of biomolecular interactions, even if it contained only article titles and abstracts.

To be able to process any text, a preprocessing pipeline was developed. For detecting protein and gene names, the BANNER named entity recognizer (Leaman and Gonzalez, 2008), known for its high performance on the standard GENETAG corpus (Tanabe et al., 2005), was chosen. BANNER is based on conditional random fields, a technique that enables straightforward detection of named entities consisting of multiple tokens. As named entity recognition is the first step in the pipeline, processing of the large PubMed dataset could be considerably optimized by limiting most subsequent processing to only those citations where BANNER detected at least one named entity, a prerequisite for detecting events with TEES.

These citations were next divided into individual sentences using the maximum-entropy based GENIA Sentence Splitter (Kazama and Tsujii, 2003a). Trained on the GENIA corpus, it is specifically aimed at processing biomedical text, achieving an F-score of 99.7% on a set of 200 unseen GENIA abstracts. To fix a few observed common error cases, a limited rule-based post-processing script was also used.

As TEES does not look for sentence-boundary crossing events, event triggers and thus events can only be detected in sentences that have at least one named entity. Therefore, we could limit the following, computationally expensive steps only to individual sentences where BANNER had detected at least one named entity. For the PubMed dataset, this meant approximately 20 million sentences.

For parsing, a setup similar to the one used to produce the supporting resources of the BioNLP'09 Shared Task was chosen. Sentences were first parsed with the BLLIP parser using David McClosky's domain-adapted biomodel (McClosky and Charniak, 2008; McClosky, 2009), a system shown to achieve the best performance on the GENIA Treebank (Tateisi et al., 2005), demonstrating applicability for biomedical text. The BLLIP parser generates PENN-style parse trees, which were further converted into dependency parses of the *collapsed-ccprocessed* style using the Stanford parser toolset (de Marneffe and Manning, 2008; de Marneffe et al., 2006).

This preprocessing pipeline has been integrated in TEES 2.0, and together with the included batch-processing system it can be used to run similar experiments on other large scale datasets.

4.1 Scaling up Text Mining

The preprocessing pipeline provided an automated system for producing the supporting data given in the BioNLP'09 Shared Task, forming the input for event extraction performed with TEES. Processing the entire PubMed dataset consumed approximately 346 CPU days, a process effectively parallelized up to 50-fold on the CSC "Murska" HP CP4000 BL ProLiant supercluster. Processing time was divided in a 1:3:1 fashion between NER, parsing and event extraction. While event extraction was much faster than parsing, it must be remembered that it also relies heavily on data generated by the parsing step.

Processing a dataset of almost 18 million citations presented several scaling issues, and provided a good test case for the adaptability of these text mining tools for real world scenarios. Most of the tools used were generally developed and tested on small datasets, such as the GENIA corpus, but the much larger PubMed dataset would contain several texts with unfore-

seen, problematic cases. We found out that e.g. the BLLIP parser would crash or never finish on only 0.09% of the sentences processed, but with the PubMed-scale dataset this minute fraction already meant 18,000 sentences. To survive situations like this, and ensure that a problem in a single sentence did not cause the whole batch to be lost, all preprocessing tools were wrapped in a layer of control code. If a tool crashed, the wrapper restarted it from the following sentence. If the tool stalled, the wrapper, tracking the appearance of tool output at a sentence level, terminated the process after a set timeout and likewise skipped the problematic sentence.

To enable parallelization, input citations were divided into small batches. A batch size of a few hundred citations allowed efficient parallelization, without wasting too much time on tool startup and shutdown times. Standard filesystems have limits on the number of files in a single directory, so batches were stored in a hierarchical directory structure.

In addition to the issues caused by tools running into unforeseen inputs, the Murska cluster could occasionally go down for maintenance breaks. More problematically, individual jobs in the cluster were processed by multiprocessor nodes with a shared memory space. Different users' processes can be allocated to the same node, and any process can accidentally consume the shared memory, causing other processes to crash. While such situations were not common, we had to take into account the fact that any processing job could fail at any time. Therefore, in addition to the tool wrappers, our large scale processing system included extensive cross-checks and methods for rerunning failed jobs.

4.2 Event Extraction Performance at PubMed-scale

Processing the PubMed dataset of 18M citations resulted in 36.5M named entities detected in 5.4M citations. 20M sentences containing at least one named entity were syntactically parsed and finally, 19.2M events were extracted.

While the pipeline of tools was known to consist of high-performance components, this performance was usually measured on small test corpora. Using random sampling, we determined whether the performance generalized to PubMed scale text mining. Sets of 100 named entities and 100 events were chosen at random and evaluated manually. In this manner, we could estimate precision, but not recall, as that would require the prohibitively slow process of fully annotating a random sample of sentences.

For named entities, we considered an entity correct if it was a cell, cellular component or any molecule taking part in biochemical interactions, including small inorganic molecules such as Ca^{2+}. Even though only proteins and genes were marked as named entities in the BioNLP'09 Shared

Task, the TEES system can detect syntactically similar relations when presented with e.g. a relation-event where the Calcium-ion is a participant. For the random sample of 100 named entities, the evaluation showed a precision of 87%, comparable to BANNER's precision of 89% on the GENETAG corpus (for an F-score of 86%).

For events, we considered the event correct if both the event and its argument entities are correct. Precision for the 100 randomly selected events was 64%, reasonably close to the 58% precision on the BioNLP'09 Shared Task corpus (for an F-score of 53%).

We also estimated precision for BioNLP'09 subtasks 2 and 3, detection of protein/gene sites of action and event modality. For subtask 2, 100 randomly selected events with subtask 2 arguments (*site* or *location*) were chosen. For subtask 3, 100 random events for which the *speculated* or *negated* modifier was predicted were chosen for both modifiers.

The subtask 2 arguments are largely external to the event and can be viewed as an attribute of the named entity participant. Therefore we evaluated them independently of the correctness of the event, determining a precision of 53%, comparable to a precision of 58% on the BioNLP'09 Shared Task development set using the same criterion of correctness.

For evaluating subtask 3, modality, of the 100 negated events 9 were incorrectly extracted to such a degree that the correctness of negation could not be determined. For the remaining 91 events, 82% were correctly marked as negated. Similarly, for the 100 speculated events 20 were too incorrect to be judged for correctness of speculation. For the remaining 80 events, 88% were correctly marked as speculated. For correctly predicted events in the BioNLP'09 Shared Task development set, precision for negation was 83% (for a recall of 53%) and precision for speculation was 77% (for a recall of 51%).

In total, these evaluations indicate that real-world event extraction performance, at least in terms of precision, seems to be consistent with measures on test corpora. Promisingly, event extraction performance was reasonably high even when the named entities are also predicted, unlike in the shared tasks where manual named entity annotations were provided.

4.3 Normalizing Events

When events are detected in text, they define the relationships between textual entities with a standardized set of argument roles and event types. However, as long as the named entities remain just strings of letters, the events are relevant only in the context of the sentence they have been detected in. Named entity normalization maps named entities, such as genes and proteins, into higher level categories, such as all synonyms of a single protein being detected as instances of the same molecule. With normaliza-

tion, events can be abstracted away from the individual sentence, showing which events refer to the same biological process. This is a basic requirement for utilizing the event data for applications such as retrieving documents describing a certain molecular process, or for joining individual events into large interaction networks.

The simplest approach to normalizing named entities is to normalize the strings by removing non-informative variation. For example, removal of capitalization, hyphens or whitespace allows detection of "actin 4", "actin-4" and "Actin4" as the same named entity. Taking advantage of the large scale of the PubMed event dataset, this approach was extended with prefix and suffix removal. For multi-token named entities, the frequency of substrings appearing on their own as named entities was used to remove non-essential prefixes and suffixes. For example, "p53" was detected as a named entity 117,000 times but "p53 protein" only 12,000 times, therefore the "protein" suffix could be removed, normalizing these into the same higher level concept "p53". On the other hand, the substring "capsid" appeared on its own as a named entity only seven times, but as part of "capsid protein" 2,000 times, indicating that this suffix should not be removed.

Using such normalization techniques, the original set of 19.2M events extracted from PubMed could be reduced to 4.5M normalized events. The most common normalized event was *Expression(Theme:"Insulin")* with 59,821 instances in the dataset. While single-argument events can be useful when searching for processes involving a specific gene or protein, generally more important are the events that describe complex relations linking together two or more genes or proteins. The most common multi-argument event was *Positive-regulation(Cause:"GNRG", Theme:Localization(Theme:"LH")* with 699 instances. This describes the process where gonadotropin-releasing hormone affects the localization of luteinizing hormone, a signaling molecule important in human reproduction.

Ultimately, string-based normalization is a very limited approach for determining the identity of biomolecular named entities. Genes and proteins can be referred to with a wide variety of synonyms that can not be detected by string normalization, such as "APO2L" and "TRAIL" for the gene "TNFSF10". On the other hand, large bioinformatics databases such as Ensembl and UniProt define unique ids for most of the known genes and proteins. Linking to such ids is important not only for normalization, but also for connecting text mining results to other bioinformatics resources.

Assigning database ids to gene and protein mentions is known as *Gene Name Normalization* and has been the subject of a task in the BioCreative III shared task (Lu et al., 2011). The GenNorm system of Wei and Kao (2011) demonstrated good performance in this task and has now been applied also to the normalization of our PubMed-scale event data in a collaboration project (Van Landeghem et al., 2013a).

In addition to normalizing to database ids, normalization methods based on gene homology have been developed by Van Landeghem et al. (2013a). Gene families often exhibit similar functionality across multiple species, making this normalization useful when events are used to extract information from literature to aid in determining the functionality of yet unknown proteins.

4.4 Applications for Events

The PubMed-scale event mining project produced a large, multi-domain dataset, potentially useful for several bioinformatics applications. The following use cases have enabled event extraction to help other bioinformatics research, and have also been important in further evaluating the extracted PubMed-scale event data.

4.4.1 The EVEX Database

Running TEES on 18M PubMed citations produced a dataset of 19.2M events. Even when converted to the BioNLP Shared Task format, removing parse and sentence splitting information, the resulting dataset takes 6.3Gb of disk space. Moreover, running experiments, even a simple search, on this dataset can be prohibitively slow, as just reading through all the files can take hours depending on the computer. As is, the dataset is thus hard to utilize for other text mining work, and certainly unusable for biologists looking for the information contained in it.

The EVEX project was founded to address these issues. First, the event data was converted into a MySQL database, allowing fast queries and further processing such as normalization, providing a basis on which to build novel text mining work (Van Landeghem et al., 2011). Second, a web interface[1] and tools like the Cytoscape plugin have been developed to provide a way for experimental biologists to use the data (Van Landeghem et al., 2012b; Hakala et al., 2012).

The EVEX database has been used in collaborative projects to provide text mining support for biological research. In a study on *E. coli* NADP(H) metabolism, EVEX has been used to extract candidate genes for regulators of this process (Kaewphan et al., 2012). By detecting candidate genes from PubMed-scale event data, text mining can save time-consuming laboratory work by selecting the most promising candidates for experimental validation. EVEX has also been applied to the study of *Arabidopsis thaliana*, combining text mining with information from experimental databases on

[1] `http://evexdb.org/`

protein–protein and regulatory interactions, building an integrative resource for domain-specific knowledge (Van Landeghem et al., 2013b).

4.4.2 Pathway Construction

Signaling pathways refer to the complex network of biochemical interactions that control the function and activities of cells. Depicted as interconnected graphs, signaling pathway diagrams, such as the ones in the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000) are used to give a detailed overview of molecular processes. Such signaling pathways are also a central tool in systems biology, the study of large-scale, complex biochemical and cellular interactions.

While constructing a signaling pathway model certainly requires much more than listing the sum of known interactions between its components, requiring interpretation and high-level filtering of information to separate the essential interactions from the superfluous ones, there has been some research into using text-mined events for automatically generating such networks.

With the PubMed-scale dataset, we visualized a subset of the original 1% dataset, focusing on interactions around interleukin-4 (Paper III). Using string-level named entity normalization, we merged individual events into a network, following the approach of Saeys et al. (2009). The event dataset covering one percent of PubMed produced a network with one major connected component consisting of 88,477 of the 232,760 nodes (38%), with the next largest connected component having only 95 nodes. We visualized the network of all proteins directly connected by an event to interleukin-4, already consisting of 19 proteins. Adding more distant connections would have increased the complexity of the network beyond easy readability.

Having finished processing the full PubMed citations dataset, we again experimented with pathway construction (Björne et al., 2010b). To visualize a manageable subset of the 19.2M events, we chose a subgraph of the well-known apoptosis signaling pathway, using the KEGG human apoptosis pathway (entry hsa04210) diagram as a template. From the KEGG pathway, we picked the interacting proteins and lists of their synonyms. From the event data, we extracted all named entities whose normalized strings corresponded to these synonyms, and visualized the network of automatically extracted events connecting them (See Figure 4.1). Due to the proteins being connected with an amount of events too large to visualize with clarity, we chose a cutoff where for each pair of proteins, 4 of that protein's most common interactions, plus 4 of the most common interactions for that protein's known KEGG interaction partners were visualized. Thus the resulting graph emphasizes that almost all of the KEGG interactions can be found in the event data (often with correct types, too), but it should also be noted

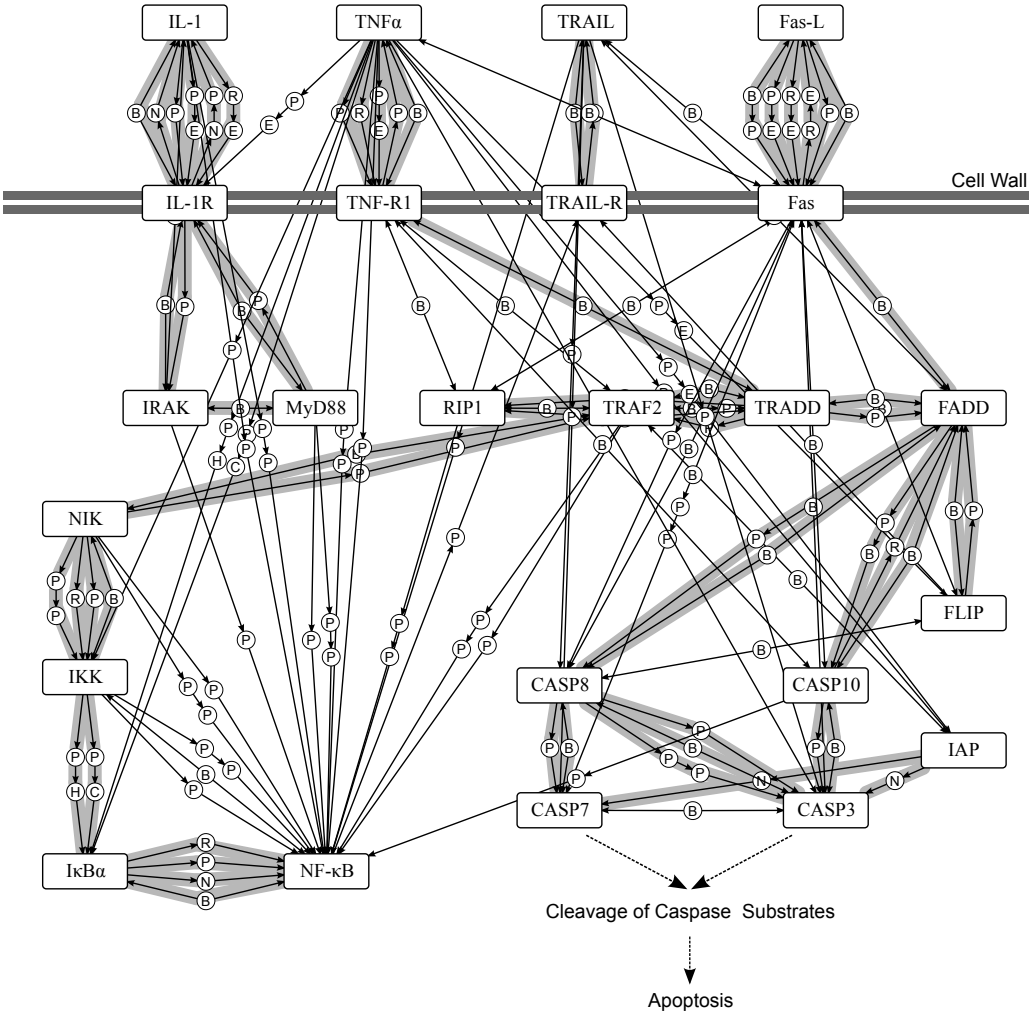


Figure 4.1: A subset of the KEGG human apoptosis pathway (entry hsa04210) reconstructed from events automatically extracted from 18M PubMed abstracts and titles. The event types are (P)ositive regulation, (N)egative regulation, (R)egulation, gene (E)xpression, (B)inding, p(H)osphorylation, (L)ocalization and protein (C)atabolism. Events corresponding to KEGG interactions are highlighted with a light grey background. The figure shows a subset of events, selected based on their frequency or correspondence to known KEGG interactions. (From Björne et al. (2010b)).

that the large number of interactions falling outside the scope of the pathway demonstrates the issues of using event data for pathway construction.

In particular, it should be noted that text mined events are not necessarily biochemical interactions. While the KEGG apoptosis signaling pathway defines the chain of detailed molecular interactions leading from TNFα binding to NF-κB, a research paper can state that TNFα positively regulates NF-κB, the most common event in the extracted pathway. Such indirect events can define higher level semantic relations, highlighting the important nodes of complex regulatory pathways. Whether these non-physical relations are useful or harmful in constructing pathways is a question that requires more research.

The event types in the apoptosis pathway demonstrate the value of a detailed event scheme. For the immediate regulation of NF-κB, we can see in the lower left corner of Figure 4.1 that IKK both phosphorylates (H) and upregulates the catabolism (C) of IκBα. This corresponds to the KEGG pathway, and the known fact that IKK (the IκB kinase) phosphorylates IκBα (the inhibitor of kappa B), leading to its degradation after detachment from NF-κB.

Without IκBα NF-κB is activated, shown also in the indirect positive regulation events connecting IKK and NF-κB, although IKK is not known to bind NF-κB as a false positive event claims. The events also show IκBα to bind and regulate (also both positively and negatively) NF-κB, which can be considered correct, as depending on its phosphorylation state IκBα either inhibits NF-κB or allows its activation by detaching from it.

The event counts for the interactions in the IKK–IκBα–NF-κB regulation system are shown in Table 4.1. Considering that events are extracted from all PubMed abstracts and titles, the numbers are perhaps surprisingly low, especially taking into account how central and much-studied the apoptosis pathway is. We have also later observed that extracted events generally provide good coverage only for the most well studied aspects of biology (Björne et al., 2012b). However, by processing also full-text articles, both the scope and nature of event coverage can be extended (Van Landeghem et al., 2013a).

Having gene names normalized to database ids makes pathway construction considerably easier. With the EVEX database, we analyzed the immediate regulatory context of the tumor suppressor protein p53 from KEGG pathway hsa04115 (See Figure 5, Van Landeghem et al. (2013a)). For this experiment, we manually evaluated the source text for the event with the highest confidence score for each interaction. All of these events are correctly extracted and correspond to the KEGG interaction. Moreover, we observed that for eight out of the nine interactions, the highest confidence event was found in a PMC full-text article, highlighting the importance of text mining access to full texts, not just titles and abstracts.

Event-based pathway construction has been studied also in the EVEX project, in construction of an *E. coli* NADP(H) metabolism regulation net-

Event	Count	Process
Positive-regulation(c:IKK, t:Catabolism(t:IκBα))	15	A
Positive-regulation(c:IKK, t:Phosphorylation(t:IκBα))	14	A
Binding(t:IκBα,t:NF-κB)	124	B
Negative-regulation(c:IκBα,t:NF-κB)	21	B
Positive-regulation(c:IκBα,t:NF-κB)	17	B
Regulation(c:IκBα,t:NF-κB)	16	B
Positive-regulation(c:IκBα,t:NF-κB)	43	C
Positive-regulation(c:IκBα,t:Positive-regulation(t:NF-κB))	28	C
Binding(t:IκBα,t:NF-κB)	29	C

Table 4.1: Event counts for the IKK–IκBα–NF-κB events shown in Figure 4.1. The events correspond to the three processes where A) IKK phosphorylates IκBα, initiating its breakdown, B) IκBα negatively regulates NF-κB by binding it and C) the overall result is positive regulation of NF-κB by IKK. Event *cause* and *theme* arguments are indicated with *c* and *t*.

work (Kaewphan et al., 2012), and in building the network-analysis tool CyEVEX (Hakala et al., 2012). Using events to connect pathways to supporting literature has also been evaluated in the PathText project (Miwa et al., 2013). Recent trends in use of text mining for biological network construction are reviewed by Li et al. (2013).

4.4.3 Protein Function Prediction

The Automated Function Prediction Special Interest Group (AFP-SIG) organizes the Critical Assessment of protein Function Annotation algorithms (CAFA) shared task. The goal of the task is to predict the function of as-yet unknown proteins, using varied bioinformatics methods, producing a set of Gene Ontology annotations. When participating in the 2010 CAFA task using machine learning methods, we also evaluated the suitability of text mined event data for protein function prediction. For our basic SVM classification model we used precalculated GO annotations produced using the Blast2GO tool (Conesa et al., 2005), provided by SIMAP (Similarity Matrix of Proteins)[2], Uniprot information on protein structures and families[3], UniGene[4] information on tissues where the protein is expressed and a feature marking whether the protein is from one of the seven CAFA target species. As text mining features, all extracted events for each protein were converted into additional features by defining all paths in such event structures leading to the leaf-node protein named entity.

[2] `http://boincsimap.org/boincsimap/`

[3] `http://www.uniprot.org/docs/similar`

[4] `http://www.ncbi.nlm.nih.gov/unigene`

With performance measured as F-score, microaveraged over the 385 predicted GO terms, our basic model achieved a performance of 52.9%. Text mining features alone had a performance of 9.4%, but combining them with the basic model reduced performance to 50.9%. The all-positive baseline for the dataset was 0.7%, and the baseline of using Blast2GO alone was 47.7%. Not too many conclusions should be drawn from these results, as our overall system performance was very low, and due to time constraints not much feature engineering was done. Regardless, the performance of text mining features alone clearly exceeds the all-positive baseline, and hints that the event data could contain information usable for computational biology tasks like protein function prediction (Björne and Salakoski, 2011a; Radivojac et al., 2013).

4.5 Open-sourcing an Event Extraction System

One of the great advantages in computer science, compared to many other fields of natural science, is the ease of sharing research. While a biochemistry laboratory might occasionally send cell lines or plasmids to a collaborating laboratory, in computer science it is possible to share all the research with everyone interested. Open sourcing research code is today easier than ever before, thanks to the internet and public project management tools developed by the open source movement. In the TEES project, it has always been the goal to produce software not just for performing a single experiment, but also for use by the larger BioNLP community.

TEES 1.0 was published on May 10th 2010, following the 2009 BioNLP Shared Task. It was released as a simple zip-archive on the project homepage. Being written largely in Python it was reasonably flexible regarding where it could be used, but was still very much lacking as generally usable research software.

TEES 1.0 was built around the concept of a procedural pipeline. In an approach somewhat similar to well-known programs such as MatLab or the R programming language, the user was expected to write simplified Python programs, defining a set of commands to perform the desired event extraction experiment. Example pipelines for the BioNLP'09 Shared Task were provided with the software, and while these supported command line parameters for training on different datasets, in practice the interface and the program were largely usable only in the specific context of the BioNLP'09 Shared Task.

Participating in the 2011 BioNLP Shared Task the goal was to generalize TEES and produce a system that would be easily applicable not only to all the BioNLP'11 tasks but also to event extraction challenges beyond this shared task. TEES being the only system to participate in all eight

BioNLP'11 tasks (and their subtasks), while also achieving best performance in four of the tasks, indicated that the program had been successfully generalized beyond the original version.

Following the BioNLP'11 Shared Task, it became again pertinent to open source the research code. However, following the 1.0 approach and just putting a zip-archive on a web page seemed limited considering the opportunities of open source, and a potential hindrance for providing ongoing support and updates. Therefore, to publish the BioNLP'11 research we decided to produce a full open source project, including code repository and history, as TEES version 2.0. GitHub[5] was chosen as a free and full-featured platform for distributing this and later versions.

The goals for the 2.0 release were to take the generalization approaches developed for event extraction in the BioNLP'11 Shared Task and further generalize them from a software engineering perspective of usability, applicability and ease of maintenance. Furthermore, the interface was to be improved and simplified, thus minimizing and later in version 2.1 often avoiding the need for TEES users to write any Python code. Finally, program usability, especially for new users, was to be improved by automating the configuration process (including that of required external programs), centralizing the multitude of configuration options and providing reasonable default settings where ever possible.

Several software toolkits exist for natural language processing, such as the Apache UIMA[6], GATE[7] and NLTK[8]. While an existing library could have reduced the need for framework code, the rapidly changing requirements of the experimental work TEES was developed for, such as the evolving event extraction tasks or our use of specialized cluster environments for large-scale processing of PubMed-data, meant that a pure Python program not requiring installation or external libraries was the safest choice for ensuring maximum flexibility.

4.5.1 Generalizing Research Code

Scientific software faces a dilemma regarding usability. In experimental research, it is highly important that the code is as flexible as possible. Being able to modify and manipulate every single step in the program is vital, so that new ideas and avenues of thought can be freely pursued without the software becoming a restriction. On the other hand, flexibility comes at a price: It tends to mean more options and settings, turning a clean interface into a jumble of choices.

[5] `http://github.com/`
[6] `http://uima.apache.org/`
[7] `http://gate.ac.uk/`
[8] `http://nltk.org/`

The TEES 1.0 pipeline files, relatively simple in the original version, had become far too complex during the development of the BioNLP'11 Shared Task. Writing a separate pipeline for each experiment proved too time consuming, introduced the possibility of further errors and most importantly reduced the ability to re-use applicable code. A famous aphorism of computer scientist David Wheeler goes: "All problems in computer science can be solved by another level of indirection." While too many levels of indirection can ultimately raise their own issues, TEES was clearly suffering from the common case of too much low level functionality being exposed on the outside. The procedural approach to defining experiments had originally been intended to give the user maximum control over the whole process of event extraction. Unfortunately it resulted in a system with dozens of internal data structures and variables managed by the user. An object oriented approach was needed to compartmentalize and modularize the code.

TEES had been written as a largely object oriented system from the beginning. It was only the outermost, user facing layer that was fully procedural, so the first step was to restructure this code, namely the pipelines, into an additional, outer level of classes managing the process of event extraction. A Detector-baseclass was developed to encapsulate the whole process of event extraction, including example generation, machine learning, evaluation and production of the final output. The object oriented interface allowed the system to hide many implementation details from the user, while still allowing detailed modification via inheritance or composition.

The object oriented interface provided the means to encapsulate overwhelming complexity in the program code, but this complexity extended also into the data files used by the system. In TEES 1.0, the system could be trained on new datasets, but to distribute the results of this training, the user was required to provide an array of files that included not only the SVM models but also the files for class and feature identifiers, not to mention the parameters required to replicate the experiment. To "encapsulate" the datafiles, a generalized model-file approach was used. Following the example of well-known machine learning systems such as LibSVM (Chang and Lin, 2011) or SVMLight (Joachims, 1999), TEES was restructured to store all the datafiles it needed in a container file, a model. Regardless of the number of machine learning systems and other steps used, training TEES 2.0 produces just this one self-contained model file which can be used to classify unannotated text. As with standard machine learning programs, the optimized classification parameters are also stored in this model. For more general settings, such as the paths to installed executable programs, the approach of the Django-project[9] was followed. Such settings are defined as variables in a user-editable Python-file, accessible for the TEES program

[9] `https://www.djangoproject.com/`

via an environment variable or a command line parameter.

In research, replicability of results is highly important. In TEES 2.0, the full standard on-screen output streams are saved in log files, producing a record of the experiments done. In recording such data, it has been our experience that it is better to simply save everything, as making a choice on the information to record can too often miss the data that turns out to be important later.

Already for the BioNLP'09 version TEES was built to take advantage of parallelization via cluster computing, but this was implemented as a system very specifically built for training the SVM classifier. In TEES 2.0, also the remote processing interface is generalized, allowing any subprocess to be transparently run on either the local machine or a remote cluster, using either simple UNIX process control tools or a job management system such as SLURM[10] (Jette et al., 2002). With this generalization, the same interface used to train classifiers in parallel was used to build a batch-processing system for processing large-scale datasets like the PubMed dataset introduced in Chapter 4.

The project to open source and generalize the TEES program shows how research questions are often instances of a more general problem. When building a solution for a particular task, it is easy to focus on the specifics and overlook the common features among related tasks. The schedules of the BioNLP'11 and BioNLP'13 Shared Tasks meant there was no time to study each task in detail, if results were to be submitted for every task. This forced the generalization of the program to enable it to be more easily applied for new contexts, but as the competitive results indicate, this turned out to be a good approach also in terms of performance.

4.5.2 TEES Use Cases

TEES 2.0 was published on August 1st, 2012. In the one year period following its publication, TEES was installed over 100 times[11]. TEES has been used for producing events for the EVEX database (Van Landeghem et al., 2011) and has been integrated in the U-Compare project, used for analyzing system combination impact on event extraction performance (Kano et al., 2011).

Several research groups have also used TEES without the direct participation of University of Turku. The *BioContext* system built on both TEES and EventMine to produce an online event database (Gerner et al., 2012). Likewise, TEES and EventMine were used to produce a dataset on human immunodeficiency virus type 1 (HIV-1) protein interactions (Jamieson et al.,

[10] `https://computing.llnl.gov/linux/slurm/`

[11] As determined by the number of downloads for the model package, downloaded during installation

2012). Neves et al. have re-trained TEES on a corpus of gene expression events relating to human embryonic and kidney stem cell research. Automatically extracted events are manually verified and used in curating the *CellFinder* database[12] (Neves et al., 2013). The NER tool *Cocoa*[13] (Ramanan and Nathan, 2013) has been integrated into the TEES preprocessing pipeline, providing an interface for processing text via the Cocoa WebAPI. The *DigSee*[14] search engine uses TEES to detect evidence sentences from MedLine abstracts through extraction of events relating genes to cancer (Kim et al., 2013a). *OncoSearch*[15] is a search engine for retrieving MedLine abstracts that describe whether a gene expression change relates to cancer progression or regression. It uses TEES to extract the gene expression change events (Lee et al., 2014). TEES has also been used in the implementation of a feature selection strategy for biomedical event extraction (Xia et al., 2014).

For the BioNLP'13 and DDIExtraction'13 Shared Tasks we used TEES to produce publicly available, pre-calculated predictions for the task datasets. This way, using TEES when participating in these shared tasks was possible also without running the program (Björne et al., 2013; Segura-Bedmar et al., 2013; Björne and Salakoski, 2013). TEES was used as a part of other entries in both the DDIExtraction'13 and BioNLP'13 shared tasks (Thomas et al., 2013; Hakala et al., 2013)

[12] `http://www.cellfinder.org`
[13] `http://npjoint.com`
[14] `http://gcancer.org/digsee`
[15] `http://oncosearch.biopathway.org`

Chapter 5

Approaches to Event and Relation Extraction

In this chapter we introduce related work relevant for the development of the Turku Event Extraction System described in chapters 2–4. Both event extraction and the related field of PPI extraction are discussed. Community-wide shared tasks have been organized for both approaches, providing a unique opportunity to evaluate the merits of different approaches in an objective and balanced setting.

5.1 Event Extraction in the BioNLP'09 Shared Task

While event-type annotations had been available for a while before it, the BioNLP'09 Shared Task was the first time event extraction became widely used in the BioNLP field. The BioNLP'09 Shared Task had 42 teams registered as participants and 24, of which two remained anonymous, submitted final results (Kim et al., 2009). The participants used a variety of parsers and other syntactic preprosessing tools. For the event extraction task itself, of the seven best performing systems, only one, the University of Concordia entry, did not use machine learning for either trigger word or argument detection. Already in the 2009 task, external resources like MetaMap, Gene Ontology and WordNet were used by 11 out of the known 22 teams, but there didn't appear to be any clear correlation between the use of such resources and system performance. Best performance was achieved by TEES with an F-score of 51.95%.

Following the Shared Task, the organizers used the final submissions to combine predictions into system ensembles. Different weighted voting schemes and numbers of systems were tested, with the best result of 55.96% (a 4 percentage point improvement over the best system) achieved by com-

bining results from the top six systems (Kim et al., 2009). The approach of system combination was further studied by Kano et al. using the UCompare framework (Kano et al., 2011) and was used in the BioNLP 2011 Shared Task by Riedel et al. (2011).

The three systems placing 2nd to 4th all reached F-scores within 2.31 percentage points of each other. The JULIE lab entry combines machine learning with a stepwise event extraction process (Buyko et al., 2009). This approach, also used by TEES, is one of the most commonly used approaches for biomedical event extraction today. The 2009 JULIE lab system used a dictionary-based approach for trigger detection and machine learning for event argument detection. Interestingly, the system also incorporates a trimming step where dependency parses are modified with a set of rules before being used as features for machine learning techniques such as the graph kernel, aiming to produce a syntactic graph closer to the semantic task of argument detection. The JULIE lab system exhibited high performance at 46.66% F-score and placed second in the 2009 task.

The University of Concordia system reached 3rd place with 44.62% F-score, and was notably the only system among the top seven to use no machine learning (Kilicoglu and Bergler, 2009). Trigger detection is based on dictionary matching, taking into account also the part-of-speech tag of the word. As with TEES, event triggers are limited to a single token in the Concordia system, but instead of always using the syntactic head token, the most semantically relevant token is used as the representation for multi-token triggers. Event argument detection is based on rules determining valid dependency paths, produced from a simplified dependency parse. The detailed analysis of syntactic structures required for defining the extraction rules of this system provides also valuable insight into the relationship between syntactic patterns and semantic relations.

Riedel et al. (2009) present a Markov logic based approach for event extraction, the "MLN(thebeast)", reaching 4th place at 44.35% F-score. This system uses a joint probabilistic model for predicting the entire event structure at the same time. In this model, e.g. the type and number of event arguments can influence the event type, in contrast to stepwise models where keywords are fully predicted before a separate event argument prediction step. Riedel et al. use Markov logic, a Statistical Relational Learning language, to define the model, allowing the authors familiar with this tool to rapidly test and develop their system. The Markov logic graph model requires the presence of all potential nodes for predicting relations between them, so, similarly to TEES graph *merging/unmerging*, proteins and event triggers are mapped to syntactic tokens and a final event network is reconstructed with a post-processing algorithm. The joint model that allows event argument predictions to affect prediction of triggers and vice versa is a promising, and still today not widely utilized approach to event extraction.

Following the BioNLP Shared Task, in 2010, Miwa et al. introduced the EventMine system, reaching a new record performance of 56.00% F-score on the BioNLP 2009 test set (Miwa et al., 2010). EventMine followed the three-step approach introduced by TEES (trigger detection, argument detection and unmerging) but implemented also the third step as a classification task. The authors evaluated five different syntactic parsers and showed that using an ensemble of parsers for feature generation can improve event extraction performance.

5.2 Event Extraction in the BioNLP'11 Shared Task

The theme of the BioNLP'11 Shared Task was generalization, expanding the event extraction approach to various new domains. In addition to the primary GENIA task, which forms the direct continuation of the 2009 task, seven additional corpora were introduced. Despite differences in annotation targets, the basic event extraction task remains the same regardless of domain. In addition to events, the 2011 shared task also included binary relation extraction tasks, the Entity Relations (REL), Bacteria Gene Renaming (REN), Bacteria Biotopes (BB) and Bacteria Gene Interactions (BI) tasks. The BioNLP'11 Shared Task had 24 groups participating, the same amount as the 2009 one (Kim et al., 2011a; Tsujii et al., 2011). The Turku Event Extraction System placed first in four out of eight tasks, the Epigenetics, Bacteria Gene Interactions, Entity Relations and Bacteria Gene Renaming tasks.

The primary GENIA task was extended by including full papers in addition to the 2009 abstracts, thus providing an important evaluation of system performance in a more realistic text mining context. Turku Event Extraction System placed 3rd in this task with an F-score of 53.30%. The best performance was achieved by team FAUST, reaching an F-score of 56.04%. Second place went to the University of Massachusetts with an F-score of 55.20% (Kim et al., 2011b).

The University of Massachusetts (UMass) entry introduced a joint model based on dual decomposition (Riedel and McCallum, 2011). While reaching 2nd place on its own, this model formed also a part of the best performing FAUST system (Riedel et al., 2011). The FAUST system combines the UMass entry with the event parsing Stanford entry. Due to the performance gap between the systems, the authors discarded voting and reranking as model combination techniques, and instead used stacking, where the predictions of the lower performance Stanford system were included in the UMass system. The Stanford system predictions are introduced as features for the UMass system, allowing the stacking system to determine the optimal use

of the stacked system. The resulting joint system demonstrated an increase of 0.84 percentage points over the UMass system used on its own. On the similar Epigenetics (EPI) and Infectious Diseases (ID) tasks, increases of 1.51 and 2.17 percentage points over UMass alone were demonstrated, respectively.

The Stanford system, used also in FAUST, shows an interesting approach of adapting the technology of dependency parsing for event extraction (McClosky et al., 2011a,b). The authors convert the event annotation to dependency trees, and use the MSTParser for predicting new ones. While the system uses a dependency parser for extracting semantic information, it also utilizes a separate syntactic parsing step (The BLLIP parser, David McClosky's biomodel and conversion to Stanford dependencies, the approach also used in TEES) for producing features for the semantic parsing step.

Expanding from the GENIA task, the BioNLP'11 Shared Task introduced two new tasks with similar corpora, the Epigenetics and the Infectious Diseases tasks (Ohta et al., 2011; Pyysalo et al., 2011a). FAUST placed first also in the Infectious Diseases task and second in the Epigenetics task. On the EPI task, TEES had a performance 18 percentage points higher than FAUST. This task specific performance difference is due to TEES being the only system that predicted additional (non-core) arguments. On the alternative "core" metric, which ignores additional arguments, TEES lead was only 0.27 percentage points.

On the ID task TEES placed 5th, 13.02 percentage points behind the task winning FAUST system. After the shared task we analyzed the results and noticed that TEES ignored ID task specific zero-argument *process* events. The four top-performing systems on the task also used the GENIA task corpus as additional training data, due to its similarity with the ID task. Implementing these approaches in TEES increased the F-score to 53.87%, a result 1.72 percentage points behind the FAUST shared task winning result. In light of these findings that show much of the large performance differences among the top systems being due to minor implementation issues, it can be said that the GENIA, Epigenetics and Infectious Diseases tasks were of roughly equal complexity, and that the top performing system for any one of these tasks could most likely be successfully adapted to a different one with limited optimization.

The Bacteria Track tasks, Bacteria Biotopes (BB) and Bacteria Gene Interactions (BI), introduced corpora more drastically different from the other main task corpora (Bossy et al., 2011; Jourde et al., 2011). These corpora were smaller, consisted largely of relations, and the BB corpus had a massive 86% of events crossing sentence boundaries, compared to less than 10% on all other corpora.

The best performing system on the Bacteria Biotopes task was by team INRA Bibliome, with an F-score of 45% (Ratkovic et al., 2011). Their Alvis

system uses the BioYatea program to extract candidate entities and assigns them to the BB task ontology. External dictionaries are used to improve term detection. Relations between entities are predicted with a rule-based system. The Alvis system also incorporates an anaphora resolution system, and the authors demonstrate that this accounts for almost 13 percentage points of their performance, due to the large number of sentence crossing relations in the BB corpus.

The supporting tasks introduced several diverse text mining targets. The Entity Relations task concerns detection of protein super- and substructures, annotated as binary relations (Pyysalo et al., 2011b). The Bacteria Gene Renaming defines also a pairwise relation extraction task (Jourde et al., 2011). The Coreference task is the most different one compared to the other tasks, concerning the detection of syntactic co-reference structures as opposed to semantic events or relations (Nguyen et al., 2011).

The best-performing system on the Coreference task was developed by the University of Utah (Kim et al., 2011c). They present an approach based on using the Reconcile system of Stoyanov et al. (2010). The machine-learning based system was not originally developed for the biomedical domain, so in the 2011 task the authors introduce several adjustments making it more suitable for biomedical text. Most importantly, a mention detector based on conditional random fields was trained on the biomedical text of the Coreference task corpus. The first place achieved with the Reconcile system, a tool that has previously achieved good results on coreference resolution, highlights the importance of using dedicated, high-performance systems for the complex syntactic task of coreference resolution also in the biomedical domain.

5.3 Event Extraction in the BioNLP'13 Shared Task

The BioNLP'13 Shared Task continued the BioNLP'11 pattern of presenting a number of structurally similar event extraction tasks that mostly differ in the domain of the annotations. Additionally, the GRN13 task results were evaluated as an interaction network and the Bacteria Biotopes subtask 1 concerned named entity recognition and categorization (Nédellec et al., 2013).

The primary GENIA task was redesigned, with the co-reference annotation included in it. The 2013 GENIA task corpus focused on full text articles (Kim et al., 2013b). The best performance in the task was achieved by team EVEX, building on the publicly available TEES 2.1 predictions (Hakala et al., 2013). The EVEX entry re-ranks the output and using a cut-off threshold aims to remove false positives, increasing performance by

0.23 pp over the second ranking TEES 2.1 baseline. The re-ranking system is machine-learning based, using support vector machines.

The third highest performance in the GE13 task was achieved by the BioSEM system of Bui et al. (2013), which extends their work published in the interim of the 2011 and 2013 BioNLP Shared Tasks (Bui and Sloot, 2012). This very interesting system achieves high performance both in terms of speed and F-score. The system is based on automatically learned syntactic patterns for events, and uses only shallow parsing. Each event in the training data is mapped to the smallest available syntactic container, such as a chunk, a phrase, or a clause. The syntax within this container is used to automatically learn a new pattern corresponding to an event annotation. When predicting new events, the system first detects potential event triggers based on a dictionary of known trigger words. Full events are then predicted by matching the syntactic pattern around the trigger against the set of patterns learned from the training data.

The BioSEM system achieved an F-score of 50.68 on the GE13 task, only 0.29 pp behind the task winning EVEX entry. Most remarkably, the system is highly computationally efficient: on a regular desktop machine, BioSEM processes the GE13 test set in 11 seconds, thus taking an average of 3.4 ms to process one sentence. By comparison, TEES or the 2011 GENIA task winning UMass system require from 1040ms to 1400ms per sentence with parsing included (Bui and Sloot, 2012).

The PC13 (Pathway Curation) and CG13 (Cancer Genetics) tasks present two new event corpora (Ohta et al., 2013; Pyysalo et al., 2013). The NaCTeM EventMine system placed first on the PC13 task (with 52.84% F-score) and second on the CG13 task (at 52.09% vs. TEES 2.1 55.41%) (Miwa and Ananiadou, 2013). EventMine was introduced in 2010 and with a pipeline approach similar to TEES it achieved the highest result on the 2009 GENIA corpus (Miwa et al., 2010). In the BioNLP'13 Shared Task the current version of EventMine was applied largely as-is to the CG13 task, but in the PC13 task used a stacking approach that utilized information from seven external corpora.

While the GRN13 task resembled the other tasks with its corpus, its evaluation was completely different: The goal was to build an overall interaction network from the extracted events and relations, and for ranking the entries the predicted networks were evaluated using the Slot Error Rate (SER) (Bossy et al., 2013a). All participants reached the <1.0 SER expected from decent predictions, and the best SER score of 0.73 was achieved by the University of Ljubljana (Zitnik et al., 2013). The system uses linear chain conditional random fields and a set of task-specific processing rules to detect events and relations from the corpus texts. The second highest SER score of 0.83 in the GRN13 task was achieved by the KU Leuven team's system, which bypasses event extraction and directly predicts with SVM:s the in-

teraction pairs used to construct the final network (Provoost and Moens, 2013).

The GRO task concerns the detection of events from a corpus with a very large number of named entity and event types derived from the Gene Regulation Ontology (Kim et al., 2013c). TEES 2.1 was the only participant in this task with a rather low F-score of 21.50% which is to be expected as the multi-class classification approach of TEES is particularly badly suited for corpora that have very many distinct types to predict.

The 2013 Bacteria Biotopes task was divided into three subtasks (Bossy et al., 2013b). Subtask 1 differs greatly from the rest of the BioNLP'13 Shared Task, as it is a named entity recognition and categorization task. Boundaries of bacteria habitat entities must be detected from text, and for each entity one or more concepts from the 1,700 term OntoBiotope ontology must be assigned. Submissions were evaluated with the SER metric, with the IRISA-TexMex system achieving the best SER score of 0.46 (Claveau, 2013). The IRISA-TexMex approach to subtask 1 has two main steps. In the first step, almost direct matches for ontology terms and terms in training data are searched for, followed by a second step where noun phrases are matched to known examples from the training data using The k-Nearest Neighbors algorithm (kNN). In subtask 2, IRISA-TexMex uses shallow linguistic information and a kNN classifier to detect relations between potential interacting pairs of named entities, reaching second place with an F-score of 40%, only 2 pp lower than the first placing TEES 2.1.

5.4 Relation Extraction

Biomedical event extraction was developed as a more detailed text mining approach building on the research on pairwise relation extraction. These formalisms have also been used in parallel, as the presence of combined event and relation annotations in several BioNLP Shared Task corpora shows. TEES is built on techniques developed in the graph kernel project, the University of Turku pairwise text mining system, and as such builds on much of the work done on pairwise text mining. Even before the graph kernel project, our first use of a biomedical text mining tool was an implementation (Pyysalo et al., 2008) of the RelEx system of Fundel et al. (2007), a purely rule-based algorithm for detecting binary relations from dependency parses.

A multitude of systems and approaches has been developed to address binary relation extraction in the past decades. In recent years, the BioCreative Shared Task has formed a similar focal point on PPI extraction as the BioNLP Shared Task on event extraction. The BioCreative tasks cover several aspects of biomedical information extraction. BioCreative II Task 3 had the topic of "Extraction of protein-protein interactions from text".

Generally, the task concerns a higher abstraction level than the BioNLP Shared Task where individual statements are recovered. The BioCreative II Task 3 defines four subtasks, the Interaction article subtask (IAS) where articles are classified as relevant for PPI or not, the Interaction pair subtask (IPS) where PPI pairs are extracted from full-text articles, the Interaction method subtask (IMS) where experimental evidence for interactions is detected and the Interaction sentences subtask (ISS) where detected PPIs are mapped to passages of up to three sentences. Of these, the last subtask, ISS, is closest to the approach of the BioNLP Shared Tasks. This task had 11 participants, with the best system mapping 19% of extracted passages to manually extracted ones (Krallinger et al., 2008).

The BioCreative III shared task also had a PPI task, consisting of two subtasks, an Article Classification Task (ACT) for detecting abstracts containing PPIs and an Interaction Method Task (IMT) for extracting experimental evidence. Eleven teams participated in the BioCreative III PPI task, with the best AUC of interpolated Precision/Recall (iP/R) curve being 68% and 53% for the ACT and IMT subtasks, respectively (Krallinger et al., 2011).

The DDIExtraction2011, First Challenge Task: Drug-Drug Interaction Extraction, was a text mining challenge for detecting mention-level drug-drug interactions. These interactions are comparable to PPI annotated in corpora such as AIMED[1], and provide an interesting point of comparison for the similar scope of the BioNLP Shared Tasks (Segura-Bedmar et al., 2011). Ten teams participated in the task, with the best result achieved by team WBI of Humboldt-Universität Berlin using an ensemble of several kernel methods with a case-based reasoning (CBR) system using a voting approach (Thomas et al., 2011).

The next DDIExtraction task was organized as part of the SemEval workshop in 2013, including also named entity recognition (Segura-Bedmar et al., 2013). For the DDIExtraction13 drug–drug interaction (DDI) extraction task, type and direction were included in the relation annotation, moving the task closer to the semantic complexity of the BioNLP Shared Tasks. Best performance for the DDI extraction task was achieved by team FBK-irst of Fondazione Bruno Kessler, who introduced a filtering step based on negation cues and named entity semantic roles to discard likely negative examples. For relation extraction the FBK-irst system uses a hybrid-kernel combining a heterogeneous feature representation, the Shallow Linguistic kernel of Giuliano et al. (2006) and the Path-enclosed Tree kernel of Moschitti (2004).

Compared to event extraction, where the complexity of the annotation often necessitates complicated multi-step machine learning setups (as de-

[1] `ftp://ftp.cs.utexas.edu/pub/mooney/bio-data/`

scribed in Chapter 2), pairwise protein–protein interaction extraction can be more easily defined as a straightforward classification task, which may have had an impact on PPI extraction systems often taking advantage of more generalized machine learning approaches. In particular, kernel methods are often used to detect the similarity between parse structures. For example tree kernels have been applied to comparison of parse trees (Zelenko et al., 2003). The graph kernel developed by Airola et al. (2008b), described in Section 3.2, continues to be utilized in PPI extraction work. The Shallow Linguistic kernel is particularly interesting as it relies on shallow parsing alone, defining bag-of-words features based on the sets of words *before/between*, *between* and *between/after* the pair of potentially interacting protein mentions (Giuliano et al., 2006). A recent overview of kernel methods used in PPI extraction is provided by Tikk et al. (2010).

5.5 Online Services

The goal in the development of many biomedical text mining systems is to provide tools to aid end users such as bioinformaticians and biologists in their everyday work. The easiest way to use such tools is as web services, and several online systems have been developed in the recent years.

The most notable is of course the web service of PubMed itself, providing a search interface to this vast database of biomedical research articles[2]. PubMed provides e.g. topic indexing via MeSH terms, synonym resolution for queries, an Advanced Search system for defining detailed queries and a subscription feature for providing updates via email on specific queries. For many biologists, PubMed is the primary, and only tool used for finding scientific articles.

Most biomedical text mining tools build on the PubMed database. The *iHOP* system hyperlinks articles via the protein and gene names that appear in them, producing a connected, navigable resource (Hoffmann and Valencia, 2004). Using Gene Ontology and MeSH terms, *GoPubMed* introduces a knowledge-based search system (Doms and Schroeder, 2005).

For detecting statements of biomolecular relations, most online systems are based on pairwise protein and gene interactions. The *MEDIE* and *InfoPubmed* systems search PubMed via subject-verb-object patterns (Ohta et al., 2006). The *Chilibot* and the *TextMed* tools use co-occurrence for retrieving pairwise relationships (Chen and Sharp, 2004; Lloyd et al., 2005). The *Ali Baba* service visualizes the search results as a graph (Palaga et al., 2009). The *STRING* database merges together many databases of biomolecular relationships, such as KEGG and UniProt, but includes also text mining,

[2] `http://www.ncbi.nlm.nih.gov/pubmed/`

bringing together a multitude of sources for biomolecular interaction data (Franceschini et al., 2013).

Following the BioNLP Shared Tasks, online services based on event extraction tools have been introduced in addition to ones based on pairwise relation extraction. The *FACTA+* system integrates event data following the BioNLP Shared Task GENIA scheme (Tsuruoka et al., 2011). The *BioContext* system uses TEES and EventMine to produce an online accessible event database, also following the GENIA event scheme (Gerner et al., 2012). The *EVEX* database, introduced in Section 4.4.1, uses TEES to produce events following the annotation schemes of the 2011 Shared Task GE, EPI and ID tasks (Van Landeghem et al., 2013a).

Chapter 6

Conclusions

6.1 Contributions of the Thesis

In this thesis we evaluated the question of extracting biomedical events defined through a complex semantic annotation scheme. A machine learning based method was developed, based on converting the event annotations into a unified graph format and decomposing the problem into consecutive, atomic classification tasks. The similarity of event annotations and syntactic dependency parses was studied in Paper I, and was later extensively utilized in developing a feature representation for event extraction relying on deep syntactic parsing.

These methods were tested in five community-wide shared tasks, the three BioNLP Shared Tasks in 2009, 2011 and 2013, and the two DDIExtraction tasks in 2011 and 2013. The methods demonstrated consistently good performance, including several first ranks achieved in each of the BioNLP Shared Tasks. The methods were implemented as the Turku Event Extraction System, a generalized, graph-based machine learning tool. The TEES system was published as a freely usable open source project, and has since then been utilized in numerous text mining projects, also by teams outside the University of Turku.

Having developed a working approach for event extraction, the suitability of the method for real-world challenges was tested by using it to extract events from the entire PubMed, the largest repository of biomedical research articles. The feasibility of such large-scale biomedical text mining was demonstrated, and the resulting dataset has become a starting point for further projects that aim to provide search solutions and apply it to bioinformatics research.

The chosen approach of performing event extraction via the three main steps of trigger detection, edge detection and unmerging is computationally intensive, especially with the added preprocessing of the chosen deep syn-

tactic parsing pipeline. Nevertheless, the excellent performance achieved in the shared tasks demonstrates the continuing validity of the approach. Applicability to the varied domain challenges introduced in the shared tasks shows the generalizability of the system, and work on the PubMed shows that the approach is also suitable for very large text mining tasks.

These being the contributions of the work, we will next discuss current issues in event extraction, and present some thoughts on possible future directions for biomedical text mining.

6.2 Ranking the Events: Relevance vs. Similarity

With the PubMed-scale event extraction project introduced in Chapter 4, event extraction is starting to become a practical tool for information extraction in the biomedical domain. In such large-scale event extraction approaches, the only metric of relevance is usually an event confidence score. A machine-learning based event extraction system can provide a confidence estimate for the events it extracts, but this confidence is not a metric of relevance, but rather of syntactic similarity with instances seen in the training data. Syntactically simple statements are naturally easier to detect and get a higher confidence score, so a statement of "actin binds profilin" will likely end up as a higher confidence event than "the nucleotide bound to actin monomers determines the affinity for profilin". However, this tells nothing about the relevance of the documents retrieved.

In 1998, Google introduced a new type of search engine based on the principle of advanced ranking of search results. Previously, internet search engines had ranked pages based on the frequency of search terms. With Google's PageRank algorithm, pages could be ranked based on how many other pages referred to them (Page et al., 1999). A page central to the concept searched (e.g. a high-quality Wikipedia page) is ranked high, as many other sources will refer to it. This provided a very insightful way of determining the *relevance* of the page. In a sense, the PageRank algorithm uses the massive, distributed, human semantic analysis of all the individual people making webpages and linking them to what they view as quality information. Since its introduction, PageRank and other methods leveraging the connections between documents have become essential tools in information retrieval.

If event extraction is to be used as an information retrieval tool, it needs to be augmented with a way to rank results by relevance. As natural language processing methods can not produce an actual understanding of the text, the simplest approach would be in the vein of algorithms such as PageRank, to utilize the human element for determining relevance. A natural source for this in scientific literature is of course the citation graph

which links together streams of research and highlights central nodes of critical importance. Unfortunately, as many aspects of scientific research, this resource is also the property of publishing companies and not available to the scientific community that created it.

The accessibility of PubMed abstracts and even full-text articles from PubMed Central has provided a rich source for event extraction. However, with advancing bioinformatics more and more biomedical research results are published and elaborated on in dedicated databases. Results of high-throughput experiments on e.g. post-translational modifications are often defined only in structured supplementary tables, data not accessible through event extraction methods (Björne et al., 2012b). Relevant information can also be found in scientific blogs or resources such as Wikipedia. Thus, processing PubMed alone will not be enough to produce a reliable gateway to biomedical information. The general web remains out of reach for all but the largest search-engine companies, and this state of affairs is not likely to change in the foreseeable future. Still, event extraction systems could already be greatly enhanced by integration of a limited number of central resources, such as UniProt and other biomedical databases in addition to PubMed.

6.3 Future Directions for Event Extraction

Event extraction was developed as a more detailed alternative for pairwise PPI extraction, accurately capturing the full semantics of the sentence. The original BioNLP'09 Shared Task defined nine event types, and subsequent tasks have introduced many more. However, this level of detail becomes also a limitation, as events can only be used to extract concepts defined in the annotation. Only the largest biomedical annotation projects such as the Unified Medical Language System (UMLS) Metathesaurus can even begin to approach universal coverage of relevant terms, and with events, not only do the terms need to be annotated, but also all the varied types of interactions between them. It's unlikely that a universal event-based approach for biomedical information retrieval will be seen, at least in the near future.

On the other hand, a simple keyword search can be used to retrieve any type of document, and when supported with algorithms like PageRank, it can also return highly relevant documents. Moreover, in the two decades since the world wide web became widely used, people have developed an instinctive understanding of how search tools work, being able to rapidly find the information they are looking for. While event extraction is thus unlikely to provide a better form of information *retrieval*, it could be very valuable in *refining* information.

With more and more information available all the time, it is not enough to simply retrieve the thousands of documents relevant for a query. More advanced systems are needed to collate and distill the essential facts from the mass of information. In practice, this approach can already be seen in the emergence of expert systems for analyzing everything from insurance claims to patient records, with the aim of providing concise, definite answers for the professional dealing with huge amounts of data. Such expert systems are developed to answer a specific question (such as analyzing the risk of certain drugs for a specific patient), an approach in line with the restricted scope of an event annotation scheme. Thus, it could be that event extraction will not provide much of an impact in terms of search systems, but could be very useful in uncovering semantic connections behind the scenes as part of a complex expert system.

On the other hand, both relation and event annotations represent ad-hoc solutions to semantic information retrieval. Inter-annotator agreement on event corpora tends to be low, and many annotations are a matter of perspective. Compare this with syntactic parsing, where linguistic theories can (in most cases) unambiguously define how a sentence should be parsed. In a sense, semantic annotation and text mining are still at a very early stage, but ultimately they might become the building blocks towards a comprehensive theory on how language conveys meaning. Such a theoretic framework may be needed to take event extraction beyond narrow domain-specific annotations and lead to more advanced and generalized information retrieval.

6.4 Final Remarks

TEES has been built from the first prototypes to rely on deep syntactic parsing, using the BLLIP parser with conversion to the Stanford dependency scheme. While high performance has been achieved with this method, we have never tested alternative methods to this computationally expensive approach. Related work, such as the Shallow Linguistic Kernel of Giuliano et al. (2006) or the template system of Bui and Sloot (2012) present intriguing cases for use of shallow parsing techniques. In future work, it would be fascinating to see more diverse use of varied syntactic NLP techniques and their applications for semantic text mining.

The relationship between event extraction and binary relation extraction in the biomedical field is an interesting one. Event extraction was originally developed to cover annotation tasks where binary relations could not fully capture the underlying semantics (e.g. situations where one protein affects a relation between two other proteins). While the current event extraction schemes can produce annotations very close to the semantics of the text,

the resulting annotations require more complex automated extraction tools. Annotating events is more time consuming, and since an event scheme is generally domain specific, also the scheme needs to be re-thought when producing a corpus for a new domain. Binary relations allow fast annotation and straightforward extraction algorithms, but at the cost of a loss of the details in the text. Which approach to use is probably largely dependent on the task at hand.

However, an interesting outcome of TEES participating in the DDIExtraction'11 task was to demonstrate that the basic feature representation of this event extraction system resulted in performance roughly at the level of high-performing binary relation extraction systems. While event extraction has more to it than detection of argument relations, on a very general level, it could be said that both event and relation extraction remain at similar levels of performance. This of course depends much on corpora used, but generally, we see F-scores in the range of 40-60%, as opposed to e.g. the 80-90% performance often seen on syntactic parsing, or the >80% performance demonstrated for semantic role labeling in the CoNLL 2009 Shared Task (Hajič et al., 2009). In terms of machine learning approaches to the problem of semantic information extraction, it seems that what ever approach or textual feature set is used, with currently available methods, biomedical text mining systems reach a similar performance level, which, while adequate for some applications, is still far from generally reliable information extraction. Does this plateau most tasks seem to hit speak of limitations on available machine learning tools, or also limitations in the way we present the language to these tools? Either way, there is still a long way to go towards effective semantic natural language processing, and it is possible that to progress, we will need entirely new ways of looking at the language.

Bibliography

Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., and Salakoski, T. (2008a). A graph kernel for protein-protein interaction extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 1–9. Association for Computational Linguistics.

Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., and Salakoski, T. (2008b). All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics*, 9(Suppl 11):S2.

Aronson, A. R. (2001). Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, pages 17–21.

Björne, J., Airola, A., Pahikkala, T., and Salakoski, T. (2011a). Drug-Drug Interaction Extraction from Biomedical Texts with SVM and RLS Classifiers. In *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction 2011*, pages 35–42. CEUR Workshop Proceedings.

Björne, J., Ginter, F., Heimonen, J., Pyysalo, S., and Salakoski, T. (2009a). Learning to extract biological event and relation graphs. *Proceedings of the 17th Nordic Conference on Computational Linguistics (NODALIDA09)*.

Björne, J., Ginter, F., Pyysalo, S., Tsujii, J., and Salakoski, T. (2010a). Complex event extraction at PubMed scale. *Bioinformatics*, 26(12):i382–i390.

Björne, J., Ginter, F., Pyysalo, S., Tsujii, J., and Salakoski, T. (2010b). Scaling up biomedical event extraction to the entire PubMed. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 28–36. Association for Computational Linguistics.

Björne, J., Ginter, F., and Salakoski, T. (2012a). University of Turku in the BioNLP'11 Shared Task. *BMC bioinformatics*, 13(Suppl 11):S4.

Björne, J., Heimonen, J., Ginter, F., Airola, A., Pahikkala, T., and Salakoski, T. (2009b). Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 10–18. Association for Computational Linguistics.

Björne, J., Heimonen, J., Ginter, F., Airola, A., Pahikkala, T., and Salakoski, T. (2011b). Extracting contextualized complex biological events with rich graph-based feature sets. *Computational Intelligence*, 27(4):541–557.

Björne, J., Kaewphan, S., and Salakoski, T. (2013). UTurku: Drug Named Entity Recognition and Drug-Drug Interaction Extraction Using SVM Classification and Domain Knowledge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 651–659, Atlanta, Georgia, USA. Association for Computational Linguistics.

Björne, J., Pyysalo, S., Ginter, F., and Salakoski, T. (2008). How Complex are Complex Protein-protein Interactions? In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM08)*, pages 125–128. Turku Centre for Computer Science (TUCS).

Björne, J. and Salakoski, T. (2011a). A Machine Learning Model and Evaluation of Text Mining for Protein Function Prediction. In *Automated Function Prediction Featuring a Critical Assessment of Function Annotations (AFP/CAFA) 2011*, pages 7–8. Automated Function Prediction – an ISMB Special Interest Group.

Björne, J. and Salakoski, T. (2011b). Generalizing biomedical event extraction. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 183–191, Portland, Oregon, USA. Association for Computational Linguistics.

Björne, J. and Salakoski, T. (2013). TEES 2.1: Automated annotation scheme learning in the BioNLP 2013 Shared Task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 16–25, Sofia, Bulgaria. Association for Computational Linguistics.

Björne, J., Van Landeghem, S., Pyysalo, S., Ohta, T., Ginter, F., Van de Peer, Y., Ananiadou, S., and Salakoski, T. (2012b). PubMed-scale event extraction for post-translational modifications, epigenetics and protein structural relations. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, pages 82–90. Association for Computational Linguistics.

Bossy, R., Bessières, P., and Nédellec, C. (2013a). BioNLP Shared Task 2013 – An overview of the Genic Regulation Network Task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 153–160, Sofia, Bulgaria. Association for Computational Linguistics.

Bossy, R., Golik, W., Ratkovic, Z., Bessières, P., and Nédellec, C. (2013b). BioNLP shared Task 2013 – An Overview of the Bacteria Biotope Task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 161–169, Sofia, Bulgaria. Association for Computational Linguistics.

Bossy, R., Jourde, J., Bessières, P., van de Guchte, M., and Nédellec, C. (2011). BioNLP Shared Task 2011 - Bacteria Biotope. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 56–64, Portland, Oregon, USA. Association for Computational Linguistics.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, ANLC '92, pages 152–155, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bui, Q.-C., Campos, D., van Mulligen, E., and Kors, J. (2013). A fast rule-based approach for biomedical event extraction. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 104–108, Sofia, Bulgaria. Association for Computational Linguistics.

Bui, Q.-C. and Sloot, P. M. (2012). A robust approach to extract biomedical events from literature. *Bioinformatics*, 28(20):2654–2661.

Bunescu, R. C. and Mooney, R. J. (2005). A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-05)*, pages 724–731, Vancouver, BC.

Buyko, E., Faessler, E., Wermter, J., and Hahn, U. (2009). Event Extraction from Trimmed Dependency Graphs. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 19–27, Boulder, Colorado. Association for Computational Linguistics.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 132–139. Morgan Kaufmann Publishers Inc.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 173–180. Association for Computational Linguistics.

Chen, H. and Sharp, B. M. (2004). Content-rich biological network constructed by mining PubMed abstracts. *BMC Bioinformatics*, 5:147.

Claveau, V. (2013). IRISA participation to BioNLP-ST13: lazy-learning and information retrieval for information extraction tasks. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 188–196, Sofia, Bulgaria. Association for Computational Linguistics.

Conesa, A., Götz, S., García-Gómez, J. M., Terol, J., Talón, M., and Robles, M. (2005). Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics*, 21(18):3674–3676.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.

Doms, A. and Schroeder, M. (2005). GoPubMed: exploring PubMed with the Gene Ontology. *Nucleic Acids Res*, 33(Web Server issue):W783–W786.

Euzéby, J. P. (1997). List of Bacterial Names with Standing in Nomenclature: a Folder Available on the Internet. *Int J Syst Bacteriol*, 47(2):590–592.

Evgeniou, T., Pontil, M., and Poggio, T. (2000). Regularization Networks and Support Vector Machines. *Advances in Computational Mathematics*, 13:1–50.

Fellbaum, C., editor (1998). *WordNet: an electronic lexical database*. MIT Press.

Floyd, R. W. (1962). Algorithm 97: Shortest Path. *Commun. ACM*, 5(6):345.

Franceschini, A., Szklarczyk, D., Frankild, S., Kuhn, M., Simonovic, M., Roth, A., Lin, J., Minguez, P., Bork, P., von Mering, C., and Jensen, L. J. (2013). STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, 41(D1):D808–D815.

Fundel, K., Küffner, R., and Zimmer, R. (2007). RelEx – Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Gärtner, T., Flach, P., and Wrobel, S. (2003). On Graph Kernels: Hardness Results and Efficient Alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop*, pages 129–143. Springer-Verlag.

Gerner, M., Sarafraz, F., Bergman, C. M., and Nenadic, G. (2012). BioContext: an integrated text mining system for large-scale extraction and contextualization of biomolecular events. *Bioinformatics*, 28(16):2154–2161.

Ginter, F., Björne, J., and Pyysalo, S. (2010). Event extraction on PubMed scale. *BMC Bioinformatics*, 11(Suppl 5):O2.

Ginter, F., Pyysalo, S., Björne, J., Heimonen, J., and Salakoski, T. (2007). BioInfer relationship annotation manual. Technical report, Technical Report 806, Turku Centre for Computer Science.

Giuliano, C., Lavelli, A., and Romano, L. (2006). Exploiting shallow linguistic information for relation extraction from biomedical literature. *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, pages 5–7.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., and Zhang, Y. (2009). The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, pages 1–18, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hakala, K., Van Landeghem, S., Kaewphan, S., Salakoski, T., Van de Peer, Y., and Ginter, F. (2012). CyEVEX: Literature-scale network integration and visualization through Cytoscape. In *Proceedings of SMBM'12, Zurich, Switzerland*, pages 91–96.

Hakala, K., Van Landeghem, S., Salakoski, T., Van de Peer, Y., and Ginter, F. (2013). EVEX in ST'13: Application of a large-scale text mining resource to event extraction and network construction. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 26–34, Sofia, Bulgaria. Association for Computational Linguistics.

Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36.

Heimonen, J., Björne, J., and Salakoski, T. (2010). Reconstruction of semantic relationships from their projections in biomolecular domain. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 108–116. Association for Computational Linguistics.

Heimonen, J., Pyysalo, S., Ginter, F., and Salakoski, T. (2008). Complex-to-Pairwise Mapping of Biological Relationships using a Semantic Network Representation. In Salakoski, T., Schuhmann, D. R., and Pyysalo, S., editors, *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008), Turku, Finland*, pages 45–52. Turku Centre for Computer Science (TUCS).

Hoffmann, R. and Valencia, A. (2004). A gene network for navigating the literature. *Nat Genet*, 36(7):664.

Jamieson, D. G., Gerner, M., Sarafraz, F., Nenadic, G., and Robertson, D. L. (2012). Towards semi-automated curation: using text mining to recreate the HIV-1, human protein interaction database. *Database*, 2012.

Jette, M. A., Yoo, A. B., and Grondona, M. (2002). SLURM: Simple Linux Utility for Resource Management. In *In Lecture Notes in Computer Science: Proceedings of Job Scheduling Strategies for Parallel Processing (JSSPP) 2003*, pages 44–60. Springer-Verlag.

Joachims, T. (1999). Making Large-Scale SVM Learning Practical. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA, USA.

Jourde, J., Manine, A.-P., Veber, P., Fort, K., Bossy, R., Alphonse, E., and Bessières, P. (2011). BioNLP Shared Task 2011 – Bacteria Gene Interactions and Renaming. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 65–73, Portland, Oregon, USA. Association for Computational Linguistics.

Kaewphan, S., Kreula, S., Van Landeghem, S., Van de Peer, Y., Jones, P. R., and Ginter, F. (2012). Integrating Large-Scale Text Mining and Co-Expression Networks: Targeting NADP(H) Metabolism in E. coli with Event Extraction. In *Proceedings of the Third Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2012)*, pages 8–15.

Kanehisa, M. and Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, 28:27–30.

Kano, Y., Björne, J., Ginter, F., Salakoski, T., Buyko, E., Hahn, U., Cohen, K. B., Verspoor, K., Roeder, C., Hunter, L. E., et al. (2011). U-

Compare bio-event meta-service: compatible BioNLP event extraction services. *BMC bioinformatics*, 12(1):481.

Kazama, J. and Tsujii, J. (2003a). Evaluation and Extension of Maximum Entropy Models with Inequality Constraints. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 137–144.

Kazama, J. and Tsujii, J. (2003b). Evaluation and Extension of Maximum Entropy Models with Inequality Constraints. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 137–144.

Kilicoglu, H. and Bergler, S. (2009). Syntactic Dependency Based Heuristics for Biological Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 119–127, Boulder, Colorado. Association for Computational Linguistics.

Kim, J., So, S., Lee, H.-J., Park, J. C., Kim, J.-j., and Lee, H. (2013a). DigSee: disease gene search engine with evidence sentences (version cancer). *Nucleic Acids Research*, 41(W1):W510–W517.

Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., and Tsujii, J. (2009). Overview of BioNLP'09 Shared Task on Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1–9, Boulder, Colorado. Association for Computational Linguistics.

Kim, J.-D., Ohta, T., and Tsujii, J. (2008). Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(1):10.

Kim, J.-D., Pyysalo, S., Ohta, T., Bossy, R., Nguyen, N., and Tsujii, J. (2011a). Overview of BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 1–6, Portland, Oregon, USA. Association for Computational Linguistics.

Kim, J.-D., Wang, Y., Takagi, T., and Yonezawa, A. (2011b). Overview of Genia Event Task in BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 7–15, Portland, Oregon, USA. Association for Computational Linguistics.

Kim, J.-D., Wang, Y., and Yasunori, Y. (2013b). The Genia Event Extraction Shared Task, 2013 Edition - Overview. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 8–15, Sofia, Bulgaria. Association for Computational Linguistics.

Kim, J.-J., Han, X., Lee, V., and Rebholz-Schuhmann, D. (2013c). GRO Task: Populating the Gene Regulation Ontology with events and relations. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 50–57, Sofia, Bulgaria. Association for Computational Linguistics.

Kim, Y., Riloff, E., and Gilbert, N. (2011c). The Taming of Reconcile as a Biomedical Coreference Resolver. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 89–93, Portland, Oregon, USA. Association for Computational Linguistics.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Krallinger, M., Leitner, F., Rodriguez-Penagos, C., Valencia, A., et al. (2008). Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome Biology*, 9(Suppl 2):S4.

Krallinger, M., Vazquez, M., Leitner, F., Salgado, D., Chatr-aryamontri, A., Winter, A., Perfetto, L., Briganti, L., Licata, L., Iannuccelli, M., Castagnoli, L., Cesareni, G., Tyers, M., Schneider, G., Rinaldi, F., Leaman, R., Gonzalez, G., Matos, S., Kim, S., Wilbur, W., Rocha, L., Shatkay, H., Tendulkar, A., Agarwal, S., Liu, F., Wang, X., Rak, R., Noto, K., Elkan, C., and Lu, Z. (2011). The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. *BMC Bioinformatics*, 12(Suppl 8):S3.

Leaman, R. and Gonzalez, G. (2008). BANNER: an executable survey of advances in biomedical named entity recognition. *Pacific Symposium on Biocomputing*, pages 652–663.

Lee, H.-J., Dang, T. C., Lee, H., and Park, J. C. (2014). OncoSearch: cancer gene search engine with literature evidence. *Nucleic Acids Research*.

Li, C., Liakata, M., and Rebholz-Schuhmann, D. (2013). Biological network extraction from scientific literature: state of the art and challenges. *Briefings in Bioinformatics*.

Lloyd, L., Kechagias, D., and Skiena, S. (2005). Lydia: A System for Large-Scale News Analysis. *12th Symp. of String Processing and Information Retrieval, (SPIRE '05), Lecture Notes in Computer Science*, 3772:161–166.

Lu, Z., Kao, H.-Y., Wei, C.-H., Huang, M., Liu, J., Kuo, C.-J., Hsu, C.-N., Tsai, R., Dai, H.-J., Okazaki, N., Cho, H.-C., Gerner, M., Solt, I.,

Agarwal, S., Liu, F., Vishnyakova, D., Ruch, P., Romacker, M., Rinaldi, F., Bhattacharya, S., Srinivasan, P., Liu, H., Torii, M., Matos, S., Campos, D., Verspoor, K., Livingston, K., and Wilbur, W. (2011). The gene normalization task in BioCreative III. *BMC Bioinformatics*, 12(Suppl 8):S2.

McClosky, D. (2009). *Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing*. PhD thesis, Department of Computer Science, Brown University.

McClosky, D. and Charniak, E. (2008). Self-Training for Biomedical Parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics - Human Language Technologies (ACL-HLT'08)*, pages 101–104.

McClosky, D., Surdeanu, M., and Manning, C. (2011a). Event Extraction as Dependency Parsing for BioNLP 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 41–45, Portland, Oregon, USA. Association for Computational Linguistics.

McClosky, D., Surdeanu, M., and Manning, C. D. (2011b). Event Extraction as Dependency Parsing. In *ACL*, pages 1626–1635.

de Marneffe, M.-C., MacCartney, B., and Manning, C. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of LREC-06*, pages 449–454.

de Marneffe, M.-C. and Manning, C. (2008). The Stanford typed dependencies representation. In *COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*.

Miwa, M. and Ananiadou, S. (2013). NaCTeM EventMine for BioNLP 2013 CG and PC tasks. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 94–98, Sofia, Bulgaria. Association for Computational Linguistics.

Miwa, M., Ohta, T., Rak, R., Rowley, A., Kell, D. B., Pyysalo, S., and Ananiadou, S. (2013). A method for integrating and ranking the evidence for biochemical pathways by mining reactions from text. *Bioinformatics*, 29(13):i44–i52.

Miwa, M., Pyysalo, S., Hara, T., and Tsujii, J. (2010). A comparative study of syntactic parsers for event extraction. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, BioNLP '10, pages 37–45, Stroudsburg, PA, USA. Association for Computational Linguistics.

Miwa, M., Thompson, P., McNaught, J., Kell, D., and Ananiadou, S. (2012). Extracting semantically enriched events from biomedical literature. *BMC Bioinformatics*, 13(1):108.

Miyao, Y., Sagae, K., Stre, R., Matsuzaki, T., and Tsujii, J. (2009). Evaluating contributions of natural language parsers to protein–protein interaction extraction. *Bioinformatics*, 25(3):394–400.

Moschitti, A. (2004). A study on convolution kernels for shallow semantic parsing. In *Proceedings of ACL04*.

Nédellec, C., Bossy, R., Kim, J.-D., Kim, J.-J., Ohta, T., Pyysalo, S., and Zweigenbaum, P. (2013). Overview of BioNLP Shared Task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7, Sofia, Bulgaria. Association for Computational Linguistics.

Neves, M., Damaschun, A., Mah, N., Lekschas, F., Seltmann, S., Stachelscheid, H., Fontaine, J.-F., Kurtz, A., and Leser, U. (2013). Preliminary evaluation of the CellFinder literature curation pipeline for gene expression in kidney cells and anatomical parts. *Database*, 2013.

Nguyen, N., Kim, J.-D., and Tsujii, J. (2011). Overview of BioNLP 2011 Protein Coreference Shared Task. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 74–82, Portland, Oregon, USA. Association for Computational Linguistics.

Ohta, T., Miyao, Y., Ninomiya, T., Tsuruoka, Y., Yakushiji, A., Masuda, K., Takeuchi, J., Yoshida, K., Hara, T., Kim, J.-D., Tateisi, Y., and Tsujii, J. (2006). An Intelligent Search Engine and GUI-based Efficient MEDLINE Search Tool Based on Deep Syntactic Parsing. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 17–20, Sydney, Australia. Association for Computational Linguistics.

Ohta, T., Pyysalo, S., Rak, R., Rowley, A., Chun, H.-W., Jung, S.-J., Choi, S.-P., Ananiadou, S., and Tsujii, J. (2013). Overview of the Pathway Curation (PC) task of BioNLP Shared Task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 67–75, Sofia, Bulgaria. Association for Computational Linguistics.

Ohta, T., Pyysalo, S., and Tsujii, J. (2011). Overview of the Epigenetics and Post-translational Modifications (EPI) task of BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 16–25, Portland, Oregon, USA. Association for Computational Linguistics.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Palaga, P., Nguyen, L., Leser, U., and Hakenberg, J. (2009). High-performance Information Extraction with AliBaba. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, pages 1140–1143, New York, NY, USA. ACM.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Provoost, T. and Moens, M.-F. (2013). Detecting Relations in the Gene Regulation Network. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 135–138, Sofia, Bulgaria. Association for Computational Linguistics.

Pyysalo, S. (2008). *A Dependency Parsing Approach to Biomedical Text Mining*. PhD thesis, Department of Computer Science, University of Turku.

Pyysalo, S., Airola, A., Heimonen, J., Björne, J., Ginter, F., and Salakoski, T. (2008). Comparative analysis of five protein-protein interaction corpora. *BMC bioinformatics*, 9(Suppl 3):S6.

Pyysalo, S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., and Salakoski, T. (2007). BioInfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, 8(1):50.

Pyysalo, S., Ohta, T., and Ananiadou, S. (2013). Overview of the Cancer Genetics (CG) task of BioNLP Shared Task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 58–66, Sofia, Bulgaria. Association for Computational Linguistics.

Pyysalo, S., Ohta, T., Rak, R., Sullivan, D., Mao, C., Wang, C., Sobral, B., Tsujii, J., and Ananiadou, S. (2011a). Overview of the Infectious Diseases (ID) task of BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 26–35, Portland, Oregon, USA. Association for Computational Linguistics.

Pyysalo, S., Ohta, T., and Tsujii, J. (2011b). Overview of the Entity Relations (REL) supporting task of BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 83–88, Portland, Oregon, USA. Association for Computational Linguistics.

Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T. W., Sokolov, A., Graim, K., Funk, C., Verspoor, K., Ben-Hur, A., Pandey, G., Yunes, J. M., Talwalkar, A. S., Repo, S., Souza, M. L., Piovesan, D., Casadio, R., Wang, Z., Cheng, J., Fang, H., Gough, J., Koskinen,

P., Törönen, P., Nokso-Koivisto, J., Holm, L., Cozzetto, D., Buchan, D. W. A., Bryson, K., Jones, D. T., Limave, B., Inamdar, H., Datta, A., Manjari, S. K., Joshi, R., Chitale, M., Kihara, D., Lisewski, A. M., Erdin, S., Venner, E., Lichtarge, O., Rentzsch, R., Yang, H., Romero, A. E., Bhat, P., Paccanaro, A., Hamp, T., Kaner, R., Seemayer, S., Vicedo, E., Schaefer, C., Achten, D., Auer, F., Boehm, A., Braun, T., Hecht, M., Heron, M., Hönigschmid, P., Hopf, T. A., Kaufmann, S., Kiening, M., Krompass, D., Landerer, C., Mahlich, Y., Roos, M., Björne, J., Salakoski, T., Wong, A., Shatkay, H., Gatzmann, F., Sommer, I., Wass, M. N., Sternberg, M. J. E., Škunca, N., Supek, F., Bošnjak, M., Panov, P., Džeroski, S., Šmuc, T., Kourmpetis, Y. A. I., van Dijk, A. D. J., ter Braak, C. J. F., Zhou, Y., Gong, Q., Dong, X., Tian, W., Falda, M., Fontana, P., Lavezzo, E., Camillo, B. D., Toppo, S., Lan, L., Djuric, N., Guo, Y., Vucetic, S. V., Bairoch, A., Linial, M., Babbitt, P. C., Brenner, S. E., Orengo, C., Rost, B., Mooney, S. D., and Friedberg, I. (2013). A Large-Scale Evaluation of Computational Protein Function Prediction. *Nature methods*, 10:221–227.

Ramanan, S. and Nathan, P. S. (2013). Adapting Cocoa, a multi-class entity detector, for the CHEMDNER task of BioCreative IV. In *BioCreative Challenge Evaluation Workshop vol. 2*, page 60.

Ratkovic, Z., Golik, W., Warnier, P., Veber, P., and Nédellec, C. (2011). BioNLP 2011 Task Bacteria Biotope – The Alvis system. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 102–111, Portland, Oregon, USA. Association for Computational Linguistics.

Riedel, S., Chun, H.-W., Takagi, T., and Tsujii, J. (2009). A Markov Logic Approach to Bio-Molecular Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 41–49, Boulder, Colorado. Association for Computational Linguistics.

Riedel, S. and McCallum, A. (2011). Robust Biomedical Event Extraction with Dual Decomposition and Minimal Domain Adaptation. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 46–50, Portland, Oregon, USA. Association for Computational Linguistics.

Riedel, S., McClosky, D., Surdeanu, M., McCallum, A., and D. Manning, C. (2011). Model Combination for Event Extraction in BioNLP 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 51–55, Portland, Oregon, USA. Association for Computational Linguistics.

Rifkin, R., Yeo, G., and Poggio, T. (2003). *Regularized Least-squares Classification*, volume 190 of *NATO Science Series III: Computer and System Sciences*, chapter 7, pages 131–154. IOS Press.

Sætre, R., Yoshida, K., Yakushiji, A., Miyao, Y., Matsubyashi, Y., and Ohta, T. (2007). AKANE system: protein-protein interaction pairs in BioCreAtIvE2 challenge, PPI-IPS subtask. In *Proceedings of the Second BioCreative Challenge Workshop*, pages 209–212. Citeseer.

Saeys, Y., Van Landeghem, S., and Van de Peer, Y. (2009). Integrated network construction using event based text mining. *Proceedings of the 3rd Machine Learning in Systems Biology workshop (MLSB)*, pages 105–14.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47.

Segura-Bedmar, I., Martínez, P., and Herrero Zazo, M. (2013). SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Segura-Bedmar, I., Martínez, P., and Sánchez-Cisneros, D. (2011). The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts. In *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction 2011: 7 Sep 2011; Huelva, Spain*, pages 1–9.

Staelin, C. (2003). Parameter selection for support vector machines. *Hewlett-Packard Company, Tech. Rep. HPL-2002-354R1.*

Stenetorp, P., Topić, G., Pyysalo, S., Ohta, T., Kim, J.-D., and Tsujii, J. (2011). BioNLP Shared Task 2011: Supporting Resources. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 112–120, Portland, Oregon, USA. Association for Computational Linguistics.

Stoyanov, V., Cardie, C., Gilbert, N., Riloff, E., Buttler, D., and Hysom, D. (2010). Coreference Resolution with Reconcile. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 156–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tanabe, L., Xie, N., Thom, L. H., Matten, W., and Wilbur, W. J. (2005). GENETAG: A tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6(Suppl. 1):S3.

Tateisi, Y., Yakushiji, A., Ohta, T., and Tsujii, J. (2005). Syntax Annotation for the GENIA corpus. In *Proceedings of the IJCNLP 2005, Companion volume*, pages 222–227.

Thomas, P., Neves, M., Rocktäschel, T., and Leser, U. (2013). WBI-DDI: Drug-Drug Interaction Extraction using Majority Voting. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 628–635, Atlanta, Georgia, USA. Association for Computational Linguistics.

Thomas, P., Neves, M., Solt, I., Tikk, D., and Leser, U. (2011). Relation Extraction for Drug-Drug Interactions using Ensemble Learning. In *Proc. of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011) at SEPLN 2011*, page 11–18, Huelva, Spain.

Thompson, P., Nawaz, R., McNaught, J., and Ananiadou, S. (2011). Enriching a biomedical event corpus with meta-knowledge annotation. *BMC Bioinformatics*, 12(1):393.

Tikk, D., Thomas, P., Palaga, P., Hakenberg, J., and Leser, U. (2010). A Comprehensive Benchmark of Kernel Methods to Extract Protein-Protein Interactions from Literature. *PLoS Comput Biol*, 6(7):e1000837.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005). Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research (JMLR)*, 6(Sep):1453–1484.

Tsujii, J., Kim, J.-D., and Pyysalo, S., editors (2011). *Proceedings of BioNLP Shared Task 2011 Workshop*. Association for Computational Linguistics, Portland, Oregon, USA.

Tsuruoka, Y., Miwa, M., Hamamoto, K., Tsujii, J., and Ananiadou, S. (2011). Discovering and visualizing indirect associations between biomedical concepts. *Bioinformatics*, 27(13):i111–i119.

Van Landeghem, S., Björne, J., Abeel, T., De Baets, B., Salakoski, T., and Van de Peer, Y. (2012a). Semantically linking molecular entities in literature through entity relationships. *BMC bioinformatics*, 13(Suppl 11):S6.

Van Landeghem, S., Björne, J., Wei, C.-H., Hakala, K., Pyysalo, S., Ananiadou, S., Kao, H.-Y., Lu, Z., Salakoski, T., Van de Peer, Y., et al. (2013a). Large-scale event extraction from literature with multi-level gene normalization. *PloS one*, 8(4):e55814.

Van Landeghem, S., De Bodt, S., Drebert, Z. J., Inzé, D., and Van de Peer, Y. (2013b). The Potential of Text Mining in Data Integration and Network Biology for Plant Research: A Case Study on Arabidopsis. *The Plant Cell*, 25(3):794–807.

Van Landeghem, S., Ginter, F., Van de Peer, Y., and Salakoski, T. (2011). EVEX: A PubMed-Scale Resource for Homology-Based Generalization of Text Mining Predictions. In *Proceedings of BioNLP'11 Workshop*, pages 28–37. Association for Computational Linguistics.

Van Landeghem, S., Hakala, K., Rönnqvist, S., Salakoski, T., Van de Peer, Y., and Ginter, F. (2012b). Exploring Biomolecular Literature with EVEX: Connecting Genes through Events, Homology and Indirect Associations. *Advances in Bioinformatics, special issue Literature-Mining Solutions for Life Science Research*.

Wei, C.-H. and Kao, H.-Y. (2011). Cross-species gene normalization by species inference. *BMC Bioinformatics*, 12(Suppl 8):S5.

Xia, J., Fang, A. C., and Zhang, X. (2014). A Novel Feature Selection Strategy for Enhanced Biomedical Event Extraction Using the Turku System. *BioMed research international*, 2014.

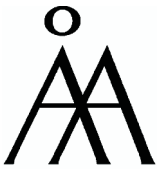
Zelenko, D., Aone, C., and Richardella, A. (2003). Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106.

Zitnik, S., Žitnik, M., Zupan, B., and Bajec, M. (2013). Extracting Gene Regulation Networks Using Linear-Chain Conditional Random Fields and Rules. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 178–187, Sofia, Bulgaria. Association for Computational Linguistics.

Turku
Centre *for*
Computer
Science

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematics



Åbo Akademi University
- Department of Information Technologies

Turku School of Economics
- Institute of Information Systems Sciences

ISBN 978-952-12-3078-3
ISSN 1239-1883