How Far Can a Few Shots Take? Exploring Few-Shot Learning in Finnish Text Classification Through Sentence Transformer Fine-Tuning

University of Turku
Department of Computing
Master's Thesis
Computer Science
May 2025
Anna Salmela

UNIVERSITY OF TURKU
Department of Computing

Anna Salmela: How Far Can a Few Shots Take? Exploring Few-Shot Learning in Finnish Text Classification Through Sentence Transformer Fine-Tuning

Master's Thesis, 53 p., 2 app. p.
Computer Science
May 2025

With natural language processing solutions on the rise, language models are getting larger with the number of parameters measured in billions, while using more and more data. In addition to this, both training a text classification model and using it later for inference can require significant computational resources. Fine-tuning language models has for a long time been a great way of adapting said models into specific domains, but they usually need significant amounts of labelled data to succeed. In this thesis, I examine the capabilities of few-shot learning by fine-tuning sentence embedding models for text classification with artificially restricted datasets created from benchmarked Finnish data to see how well considerably lighter models with fewer data perform compared to state-of-the-art solutions.

As the main method, this thesis explores few-shot learning by using the SetFit library as a way to fine-tune sentence embedding models for text classification. SetFit enables the use of extremely small datasets for training, and dataset sizes of 8, 16, 32 and 64 samples per label are tested. The analysis includes comparing the results from several fine-tuned models, including both monolingual and multilingual sentence embedding models, with varying tasks: multilabel register (genre) classification, multilabel toxicity detection, multiclass news category classification and multiclass discussion forum topic classification.

Even though state-of-the-art results are not reached by fine-tuning sentence embedding models, SetFit shows promise especially in the multiclass prediction tasks. While the benchmark results are higher, SetFit achieves decent model performance with smaller datasets. In some cases, it looks like 32 or even 16 examples per label might be enough to get the most out of this method. From the different sentence embedding models tested, the 125M parameter monolingual Finnish one fares the best in all tasks when fine-tuned with SetFit.

The results of this thesis are promising for use cases where the amount of data and computational resources are limited. To my knowledge, this is the first time SetFit has been studied with Finnish data. Previously, Finnish few-shot classification has been tested with the aid of large language models, thus requiring significant computational resources. Compared to these methods, SetFit is very light to use and could lower the experimentation threshold for text classification tasks.

Keywords: natural language processing, few-shot learning, text classification, SetFit, sentence embeddings, Sentence Transformers

Contents

1 Introduction

In recent years, the field of natural language processing (NLP) has seen a lot of development in numerous areas. Recently, as a backlash to the ever-growing resource needs of large language models, smaller and more efficient foundational model architectures have gained ground [1]. Still, in the centre of all language model training remains the question of available data, which can be difficult to source especially for lower resource languages or data-sparse domains. In this thesis, I will examine how well considerably lighter models with fewer data perform compared to state-of-the-art solutions in Finnish text classification.

Different kinds of text classification problems are at the core of NLP research. Fine-tuning foundational Transformer models has proved a powerful solution for many domains, and in some cases has already reached near human-level performance [2], [3]. However, Transformer model fine-tuning and inference take up considerable computational resources that might not be readily available for all. End-users are often faced with balancing computational capacities and data security with model accuracy, since most existing solutions require significant computational resources, and the models are often trained and used for inference in the public cloud [4]–[6]. Lighter models that can be easily run for inference on-device might be useful for cases where high privacy is necessary, or if one desires to restrict resource consumption for sustainability reasons.

In addition to computational resources, most language models require a lot of

data even for fine-tuning. This already might create a problem in data-sparse languages and domains. The limited amount of data might be due to copyrights and other use restrictions, e.g. for data privacy reasons in the medical field, bad or noisy quality of data parsed from the Internet, or there just might be limited amounts of data in general in the designated domain. Even if there are available data resources, the annotation costs can rise high. Although current trends seem to favour the amount of data for the quality of data, research supports that in order to get high-quality results, one must use high-quality data that is suitable for the task and domain at hand [7]–[12]. Methods that require fewer data and computational resources could possibly lower the AI experimentation and adoption threshold.

If data availability is a problem, what if there was a way to fine-tune language models with fewer data? These kinds of few-shot learning methods do exist, but many rely on large language models (LLMs) and thus do not answer the question of limited computational resources, as they are quite expensive when used for inference. The Hugging Face library SetFit addresses both issues by harnessing the capabilities of Sentence Transformers and contrastive learning and proposes a method for fine-tuning Sentence Transformers for text classification purposes with as few as eight examples per label [13]. They also report success in multilingual experiments, which is echoed by many others [7], [14]–[17].

The issue with Finnish NLP is that there aren't many labelled datasets, and only a few reported benchmarks for text classification. There are some reported benchmarks [2], [9], [18], and some datasets have been created by machine-translating labelled English data [12], [18]. There has been success in text classification in Finnish with fine-tuning both monolingual Finnish base models [2], [12] as well as multilingual models [9]. These scenarios, however, have utilised large datasets, and there are very few examples of research on few-shot text classification in Finnish. The existing ones have used LLMs for in-context learning [3], [18]. Fine-tuning Sen-

tence Transformers for Finnish few-shot text classification has not been researched at all.

The goal of this thesis is to benchmark the SetFit few-shot fine-tuning method for Finnish text classification, as well as understand which kinds of tasks it is the most suitable for. As this research will be the first ever reported benchmark for fine-tuning Sentence Transformers for few-shot classification in Finnish with SetFit, it will provide insights on how the method works with different kinds of Sentence Transformer models and data formulations. It will also add into the existing pool of Finnish few-shot experimentation. If the method works, it will lower the threshold for NLP experimentation in lower end devices and data-sparse domains even in Finnish.

With these goals in mind, I will conduct the research by focusing on the following research questions:

1. How well do the fine-tuned Sentence Transformer models perform with Finnish tasks compared to the reported benchmarks? Can the SetFit method reach results comparable to the state-of-the-art?

2. Is there a difference in performance when fine-tuning Finnish Sentence Transformer models versus multilingual Sentence Transformer models with Finnish data?

3. Is there a difference in performance when fine-tuning Sentence Transformers with multilabel classification versus multiclass classification in Finnish?

My hypothesis regarding the first research question is that even though SetFit seems like a promising solution, I doubt that it will reach state-of-the-art results. This hypothesis is backed up by the results from the original authors [13]. It remains to be seen how much the SetFit method's results differ from the state-of-the-art. If the difference is minimal, SetFit could offer a good alternative for text classification

fine-tuning as it enables faster training with fewer data.

Regarding the second research question, there are successes in Finnish text classification in favour of both monolingual [2], [12] and multilingual models [9], [19], although the differences between the two methods have not been too large. Therefore, I do not have a clear hypothesis for this research question.

For the third research question, my hypothesis is that multiclass classification tasks with one class per example will fare better than multilabel tasks, where one example might belong to many different classes. This is due to SetFit's fine-tuning process, which has the purpose of distancing the different classes in the embedding vector space [13]. This should be easier if the classes are distinct from one another.

This thesis will be structured in the following manner:

Chapter 2 Theoretical Background: I will introduce the main concepts used in this thesis. These include machine learning, language models, transfer learning, word and sentence embedding representations, text classification and few-shot learning as well as introducing the main method for experiments conducted in this research, the Hugging Face library SetFit.

Chapter 3 Methodology: I will introduce the data used in the experiments as well as the experimental setup, including the task definition and outline, as well as the evaluation methods used to compare the SetFit performance to the original benchmarks.

Chapter 4 Results: In this chapter, I will go through the classification results of the fine-tuned Sentence Transformer models as well as look deeper into the learning process when fine-tuning models with SetFit. I will compare different sizes of datasets with all the tasks and models I have chosen to test.

Chapter 5 Discussion: I will gather my findings and draw up some generalisations based on the results presented in Chapter 4. In addition, I will examine the results of the analysis process and the observations that I have made based on it.

This chapter will also include any limitations or problems that have arisen during the research.

Chapter 6 Conclusion: This chapter will sum up the thesis and introduce ideas for further research.

2 Theoretical Background

In this chapter, I will introduce the idea behind modern neural network architectures and take a look at utilising neural networks in natural language processing in the sense of language models and transfer learning. I will also touch on the special case of few-shot learning when fine-tuning pretrained language models.

The first steps in neural network research were taken already in 1943 by McCulloch and Pitts [20] with the idea of modelling the neural activity of the human brain. However, the first artificial neural network was created in 1958 by Rosenblatt [21]. This network model is called a perceptron, and it consists of a single layer that imitates the behaviour of a neuron. A perceptron calculated a weighted sum of the inputs fed to it, and feeds it through a threshold function before outputting a result, such as probabilities for different classes in a classification task. Even though Rosenblatt's invention holds only one hidden layer, perceptrons can be stacked in order to create a multilayer perceptron with more than one hidden layer between the input and output layers. This kind of a multi-layer network is often called a deep neural network. Perceptron's simple architecture is the basis of modern neural network solutions.

When speaking of machine learning, it is sensible to make the distinction between supervised and unsupervised learning. Supervised learning has a certain predefined goal, be it a correct translation, an accurate audio transcript or correctly predicted labels in a classification task. In unsupervised learning, such truth does not exist,

but the machine is left to find patterns in the given training data on its own. An example of an unsupervised machine learning task is topic modelling. [22] This thesis will focus on supervised machine learning, although sentence transformers are suitable for unsupervised tasks as well [23]. However, the tasks chosen to evaluate the performance of few-shot learning will be classification tasks, which are supervised by nature.

2.1 Language Models and Transfer Learning

Language models are an essential part of natural language processing. Training a machine learning model from scratch every time you need to change or update your task is not usually a great idea, as training a well-performing model often requires considerable computational resources, data, and time. Often, the smart solution is to utilise previous knowledge in the form of pre-trained models. In the context of NLP, these pre-trained models are called language models, which calculate probability distributions of words in a language.

There are several different types of architecture and levels of complexity for language models. At its simplest, a language model may refer to an n-gram model with likelihoods of n-length sequences of words, for example, a bi-gram model calculates the probability of sequences of two consecutive words. This type of language model is purely statistical and does not take the surrounding context into account [22]. Neural language models were created as a context-aware alternative for statistical language models. Examples of neural models are Recurrent Neural Networks, which introduced a hidden state to the hidden layer for memory reasons [24] and Transformers [25], with the introduction of the attention mechanism. On the more complex end of the spectrum are general-purpose large language models with billions of parameters, such as GPT [26]–[28], BLOOM [29] and LLaMA [30]. The field of language models is rapidly evolving, and new types of breakthroughs are

continuously introduced.

Language models can be fine-tuned for specific tasks while preserving the previous knowledge of a language the model has already learnt. This is called transfer learning. In order to utilise transfer learning, you must first choose a base model for fine-tuning. Then you need to fine-tune your model with data specifically curated for the task at hand. A language model can be fine tuned for text classification (such as toxicity detection, register classification and spam detection), part-of-speech tagging, named entity recognition, and many other tasks. A model will only learn from the data provided to it, and it will not have capability to apply the information in ways unknown to it.

A pitfall that one might encounter while fine-tuning for a specific task is overfitting: a phenomenon where a model adapts too closely to the target task and the training data it has received [22]. This is also known as catastrophic forgetting, as it essentially means that the model forgets previously learnt information to replace it with the new domain knowledge. Overfitting leads to models that do not perform well with unseen data. There are several ways to prevent overfitting, including using a designated validation set for parameter tuning in addition to training and test datasets, using cross-validation or a dropout system, where randomised units are dropped from the neural network during training [22].

Even though there are some datasets for training machine learning models for specific tasks, Finnish language model resources are, however, somewhat limited. There are some monolingual Finnish general-purpose language models such as FinBERT [2], used previously in e.g. [9], [12], and a Finnish version of a sentence BERT, Finnish Paraphrase [31], which is trained from the basis of FinBERT. Another solution with more options is to try fine-tuning a multilingual model such as XLM-RoBERTa [32] or multilingual M-BERT [33]. Large language models also provide multilingual opportunities, but in this thesis, I will focus on smaller-scale

Transformer models.

According to Rönnqvist et al. [19], multilingual models might have an advantage to monolingual models, although Eskelinen et al. [12] have proved that with sufficient amount of data, fine-tuned monolingual models outperform multilingual models in monolingual tasks. In this thesis, I will utilise transfer learning by fine-tuning existing sentence embedding models for certain classification tasks. I will compare the performance of both monolingual Finnish as well as multilingual embedding models with the reported state-of-the-art benchmarks that exist at the time of writing. With the constant advances in the field of NLP, there might already be solutions that outperform the benchmarks that I have used in this thesis.

2.2 Word and Sentence Embeddings

"Embedding" is an umbrella term to describe numerical representation of language, images, audio, and data in general: essentially, an embedding is a vector. In the context of natural language processing (NLP), embeddings most often refer to words, sentences, or texts mapped into a vector space with predefined dimensions [22]. Embeddings derived from text can be used to measure similarity between different words or documents, text classification, document clustering or grouping, or feature extraction for further NLP tasks.

The notion of modelling the language with vectorised words or sentences is nothing new. Most language models use vocabularies that map words into numbers. A similar thought can be applied to word and sentence embeddings, although they also model semantical relationships within the language. A famous example of this is the example of word vectors "King" - "Man" + "Woman" that result in "Queen", introduced by Mikolov et al. in 2013 [34], also visualised in Figure 2.1. With sentence embedding models, we are able to retain more context and even complex semantical structures.

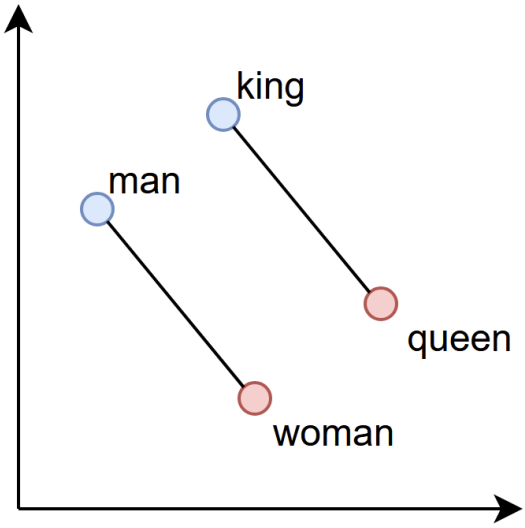

Figure 2.1: Representation of words in a vector space

Embedding models are language models aimed at grouping semantically similar words or sentences closer together in the vector space, and respectively creating distance between semantically distant words or sentences [13]. The output of an embedding model is a vector, a dense numerical representation of text that can be used in downstream machine learning tasks. In this section, I will briefly introduce some of the most influential advances. The models are often validated using a similarity metric to measure the distance in the vector space by calculating either the angle or the distance between the embedding vectors being compared [35].

2.2.1 Word Embeddings

As mentioned above, Mikolov et al. [34] introduced the idea of capturing semantical relations between word vectors, which they refer to as analogies. Their Word2vec models improved the results from previous neural network architectures, using unsupervised CBOW and Skip-gram methods to create vector representations of words from large datasets. These models were evaluated by semantic and syntactic word similarity tasks, and perform especially well compared to previous models in seman-

tic tasks while being much more computationally efficient.

Global Vectors (GloVe) [35] combine the local context implementation used in the Word2vec Skip-gram model with global matrix factorisation. Pennington et al. criticise the two methods for poor generalisability: they either fail with statistical tasks or semantic analogy tasks, while performing well in the other. As a solution for this problem, they propose an unsupervised global log-bilinear regression model architecture that utilises word co-occurrences and captures global corpus statistics. The resulting model is somewhat lighter than Word2vec models and outperforms them in accuracy in word similarity, word analogy and named entity recognition tasks.

The aforementioned models create an embedding vector per each word form, which results in several vectors per word in morphologically complex languages. FastText [36] takes word morphology into account, and instead of assigning each word a distinct vector, they propose a model where a word is represented as a sum of character n-gram vectors, an extension of the Skip-gram model introduced in [34] and outperforming the previous implementation. Languages such as Finnish with its 15 cases for nouns can benefit from models that take the word morphology into account, since some variations of a word might appear rarely or not at all in a training corpus. Because of the character-level representation of words, the FastText model is able to use data more robustly and thus requires fewer data and can better model infrequent words of even out-of-vocabulary words.

2.2.2 Sentence Embeddings

In order to get sentence embedding vectors, word embedding vectors had been previously mapped to sentences. Skip-Thought [37] applies the idea of unsupervised word embedding model training to sentences with a modified Skip-gram implementation: instead of using a word to predict surrounding words, they tried predicting sur-

rounding sentences from a given sentence. This performed better than the previous implementations that had been utilising word embeddings to map sentences.

Conneau et al. and their InferSent model [38] provide a supervised alternative for sentence embedding retrieval. They compare multiple neural network architectures and propose a bi-directional long short-term memory (LSTM) model as a solution for retrieving sentence vectors. As training data, they used the SNLI corpus [39] with sentence pairs labelled with "entailment", "contradiction" and "neutral" tags. They found that sentence embeddings work well in transfer learning tasks, and that the proposed solution is more computationally efficient compared to the unsupervised methods.

The SNLI data used in InferSent training is also used in the training process for Universal Sentence Encoder [40]. Cer et al. found that the transformer architecture [25] is optimal for training sentence embedding models for transfer learning. They evaluated the model by providing embeddings from the models to task-specific deep neural networks. They also tested training a deep averaging network architecture where input embeddings are averaged before passing through the network; this implementation is computationally cheaper but not as well performing as the Transformer-based models.

Although Transformer language models such as BERT [33] and RoBERTa [41] can be used to derive sentence embeddings by, for example, averaging the last hidden layer into a fixed size vector, the results are usually worse than even basic word embedding models such as glove [23]. Sentence Transformers (SBERT) [23] is based on the BERT architecture that uses Siamese networks to derive sentence embeddings while retaining semantical meaning. This means that the model is fine-tuned by comparing the pooling outputs of two sentences and updating the model's weights to contain the semantical meaning. A visualisation of the model architecture can be viewed in Figure 2.2.

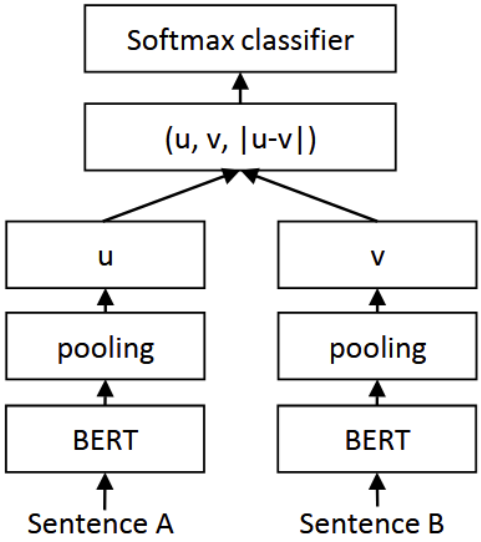

Figure 2.2: SBERT architecture [23]

SBERT was a significant improvement from earlier sentence embedding models and compared to the original BERT, the computational cost is very low when computing an embedding for a sentence [23]. This is important especially when trying to find the most similar sentence pairs and the dataset is large, since embedding similarity must be calculated for each sentence pair. The model is fine-tuned with Natural Language Inference (NLI) datasets SNLI [39] and MNLI [42] and validated with both supervised and unsupervised tasks, including fine-tuning the SBERT model with regression objective function. Contrary to [38] and [40], Reimers et al. [23] do not recommend SBERT for transfer learning, even though it surpasses the previous sentence embedding models in transfer learning tasks. Instead, they suggest that the original BERT should be used for the purpose.

NLI datasets have been a solid base for Sentence Transformer training. These kinds of datasets are annotated with the information if two phrases contain positive, negative or neutral textual entailment [39]. In addition to NLI datasets, other common ways to train a Sentence Transformer are by using paraphrase or machine translation datasets [31].

2.3 Text Classification and Few-Shot Learning

Text classification is a fundamental NLP task, where a given text sample is assigned one or multiple predefined classes. Different classification tasks can be divided in binary classification with two available classes, such as spam detection (spam or not spam), multiclass classification with more than two available classes, such as sentiment detection (positive, negative, or neutral sentiment), and multilabel classification, such as emotion detection where each sample can be assigned with several coexisting classes [22]. Text classification can be approached with more traditional supervised machine learning methods, such as support vector machines or naive Bayes algorithms [22], but recently, the state-of-the-art results in classification tasks have been achieved with fine-tuned Transformer models [9], [12]. Although LLMs can also be used for zero-shot and few-shot text classification tasks, smaller models fine-tuned with larger datasets still often achieve the best results [43].

Traditional classification tasks may require large amounts of data in order to train a well-performing machine learning model. Textual datasets are often imbalanced, and it is not always possible to acquire an adequate number of data points in order to teach the model to reliably recognise the correct class, be it due to issues in sampling or the cost of annotations. This can be the case in online register classification [9], [19], [44], toxicity and cyber harassment detection [12], [45] as well as different tagging tasks such as part-of-speech tagging and named entity recognition [46] among others. When a dataset is imbalanced with classes that are under-represented, the model will be biased towards the major classes in the data set. Optimising the overall accuracy of the model might lead to ignoring the minor classes. Strategies for dealing with imbalanced datasets include, for example, data augmentation for under-represented classes with generative language models [47], using a different loss algorithm (dice-loss vs binary cross-entropy) [46] and converting classification tasks into entailment prediction tasks [48].

Few-shot learning approaches are options to consider when the data amount is limited or when the data contains classes with few examples [13]. "Shots" are used to describe the number of input-output pair examples introduced in training the model, thus few-shot means training a model to perform a task with only a few examples of data points. Another case is zero-shot classification, where the model is used to predict cases in unseen domains or languages. This could for example mean natural language inference (NLI) tasks [39], [42], [49], or data in a language that has not been used in training or fine-tuning the model used for predictions [19].

When you have some labelled data, there are several ways you can approach your classification problem. Often, a simple solution could be to use your small dataset to fine-tune a pre-trained language model such as RoBERTa [41] or BERT [33]. However, if your dataset is minimal or very imbalanced, achieving good classification results might not be possible [11], [13], [15]. You could also try in-context learning, parameter-efficient fine-tuning or pattern exploiting training [50]. According to Tunstall et al. [13], these scenarios can be impractical, as they often rely on large language models with billions of parameters, such as GPT-3 [26] and GPT-4 [28]. This means that the required computing resources are significant and not accessible for everyone. In addition, parameter-efficient fine-tuning and pattern exploiting training tasks require manually generated prompts and are thus dependent on the quality of the prompt-engineering.

Despite previous successes in Finnish NLP [2], [9], [12], [18], [31], [51]–[53], there is very little research on few-shot text classification in Finnish. The existing research is limited to Kortesalmi's [3] comparison of LLM-reliant in-context learning method to traditional machine learning algorithms and a fine-tuned Transformer model, and a few-shot evaluation of the Finnish GPT (FinGPT) presented by Luukkonen et al. [18]. One of the goals of this thesis is to explore this area of Finnish NLP and add to the existing research.

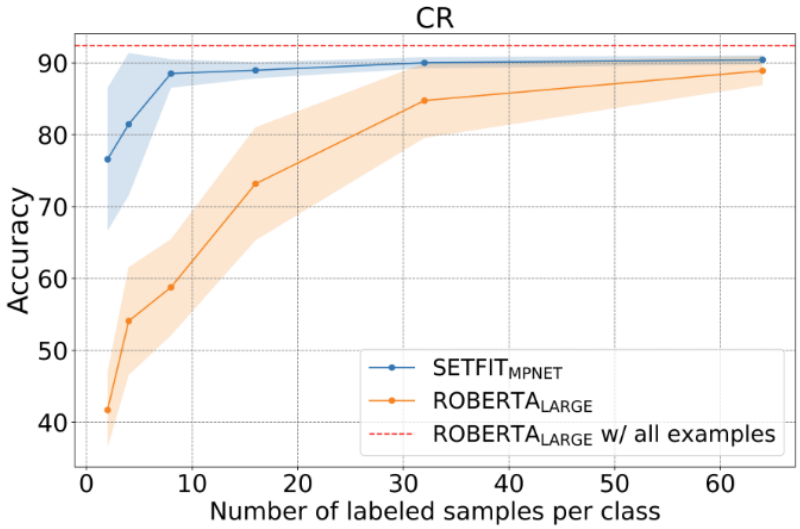

Figure 2.3: SetFit performance compared to fine-tuned RoBERTa-large in customer review sentiment classification [13]

2.3.1 SetFit

As mentioned above, many few-shot solutions rely on large language models and are thus quite resource heavy. But what if you are in a situation where you lack computational resources in addition to data?

Tunstall et al. propose a solution to a data-sparse task with SetFit [13], a few-shot fine-tuning framework based on Sentence Transformers [23], an architecture not originally intended for transfer learning. Sentence Transformers have previously been used for text classification by applying a logistic regression function on top of the sentence embeddings retrieved from the model [54], [55], but SetFit introduces the idea of fine-tuning any given Sentence Transformer model in a Siamese manner with the classification task in mind. Because SetFit supports any Sentence Transformer, it provides multilingual support if the user wants to use a multilingual model. On top of all this, Tunstall et al. claim that only 8 training examples are needed for a fine-tuned SetFit model to perform at a competitive level compared to a RoBERTa-large fine-tuned with the full dataset.

SetFit's performance is evaluated against a fine-tuned RoBERTa-large and sev-

eral other few-shot methods: pattern exploiting training (PET) based ADAPET [56], Perfect [57], and parameter-efficient fine-tuning (PEFT) based T-Few [58] in several test datasets, including emotion recognition, sentiment detection, spam detection and news topic detection [13]. The SetFit method on average outperforms both ADAPET and Perfect methods and is comparable to the T-Few 3B parameter model with n = 8 examples while being 27 times smaller, and ends up outperforming it with n = 64 examples [13]. Figure 2.3 displays the performance against a fine-tuned RoBERTa: while never quite reaching the accuracy achieved by a RoBERTa model fine-tuned with the full customer review sentiment dataset, the SetFit fine-tuned MPNet model outperforms the RoBERTa one with smaller amounts of data.

Compared to the T-FEW method that performs on a similar level, SetFit is faster in both training and inference [13]. With n = 8 examples per label, SetFit takes about 30 seconds to train. T-FEW takes over 20 times longer than that and requires more GPU memory to do it. In addition to this, SetFit needs significantly smaller disk storage space: 163 to 26 times less than T-FEW with the tested models [13]. These factors add to the attractiveness of the SetFit method for real-world applications.

Multilingual Experiments

Tunstall et al. also perform multilingual experiments with a multilingual MPNet model [1] compared to a cross-lingual XLM-RoBERTA-base[2] and ADAPET. The evaluation is done with Multilingual Amazon Reviews Corpus, which contains reviews in English, Japanese, German, French, Spanish, and Chinese [59]. SetFit outperforms both methods with limited data, but the performance is weaker than that of the XLM-RoBERTa model, which is fine-tuned with the full dataset [13].

SetFit has been used successfully in many low-resource language research tasks.

[1] huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

[2] huggingface.co/xlm-roberta-base

These include offensive content detection in Tamil [16], radiological text classification in Danish [17], few-shot classification benchmarking in Polish [15], legal judgment prediction in Korean [14], and Arabian dialect sentiment analysis [60]. It also shows promise in Dutch protest tweet classification [7], legal text classification [11] and financial domain text classification [8]. Even though certain experiments are done with a larger number of data samples [7], [14], [16], [17], [60], performance comparable to fine-tuned Transformers or other few-shot methods is achieved in true few-shot settings as well [8], [11], [15]. SetFit even sometimes outperforms a fine-tuned Transformers model [11], [16]. In [15], SetFit comes in second place when benchmarking few-shot methods for various classification tasks in Polish but achieves significantly lower results than in-context learning with GPT-3.5. Loukas et al. [8] found that by combining SetFit with representative samples chosen by a human expert, they were able to surpass state-of-the-art results in the financial domain.

There is no previous research or benchmarking for SetFit in Finnish few-shot classification. The multilingual benchmarks do not include Finnish or other languages from the Finno-Ugric language family in their datasets, even though the multilingual Sentence Transformer model used in the experiments in [13] does include Finnish in its training data [61]. This thesis aims to expand this field of research into Finnish language tasks to gain some knowledge of the suitability of few-shot classification and SetFit in particular that could prove useful for a relatively low-resource language.

Training

SetFit training process is divided in two steps and described as follows [13]:

1. Fine-tune a Sentence Transformer with contrastive sentence pairs in a Siamese manner

2. Train a classification head with data that is generated by the Sentence Trans-

former that is fine-tuned in the first step

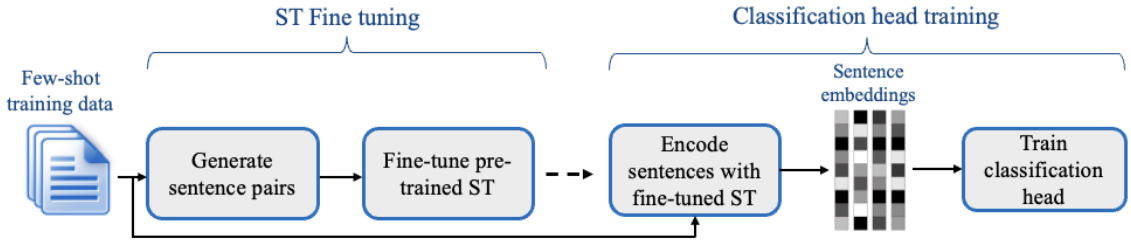

Figure 2.4: SetFit fine-tuning process [13]

Figure 2.4 illustrates the steps of the Sentence Transformer fine-tuning and classification head training. In the first step, the Sentence Transformer model is fine-tuned in a Siamese manner. This is achieved by essentially augmenting the data by contrasting pairs and generating sets of positive (randomly chosen examples with same label) and negative (randomly chosen examples with different label) pairs for each label in the data set.

In the fine-tuning phase, the model will generate embeddings for each pair it receives and will modify the weights of these examples depending on the pair quality: positive pairs will receive more similar embeddings and negative pairs more differing embeddings. This contrastive training method enables the use of small datasets, since it compares pairs instead of individual examples. For example, a classification task with eight examples from two discrete classes each will result in $\binom{8}{2} * 2 = 56$ positive pairs and $8*8 = 64$ negative pairs, augmenting the dataset from 16 examples up to 120 unique pairs. The number of pairs grows exponentially to the number of examples and classes.

After fine-tuning the Sentence Transformer model, a classification head is trained on top of it. The training data is encoded by the fine-tuned Sentence Transformer, and the resulting embeddings as well as the labels form the training set. In [13], they use a logistic regression model to fine-tune the classification head. Since few-shot

training might be unstable [62], [63], the authors created 10 random train splits in order to estimate the model's performance.

3 Methodology

In this chapter, I will introduce the data used in the study, as well as the corresponding benchmarks achieved with said data. Following that, I will outline the experimental setup of the thesis.

3.1 Data and Classification Tasks

For the experiments, I will use pre-existing, monolingual Finnish text classification corpora with reported fine-tuning benchmarks. I have chosen both mono-label and multilabel corpora, with the "simplest" corpora containing only 10 unique classes [64], [65], and the most complex one 39 different classes with the possibility of multiple classes per example [9]. Another variable in the corpora is the style of language, which varies from the more formal news-texts [9], [65] to informal or straight offensive tone of language [9], [12], [64]. Some of the corpora also contain machine translations [9], [12] of varying quality and transcripts from spoken language [9].

3.1.1 FinCORE

FinCORE corpus[1] consists of 10 754 Finnish web-crawled texts labelled into nine main registers and 30 subregisters (genres) [9]. One text can have more than one

[1] https://github.com/TurkuNLP/FinCORE_full

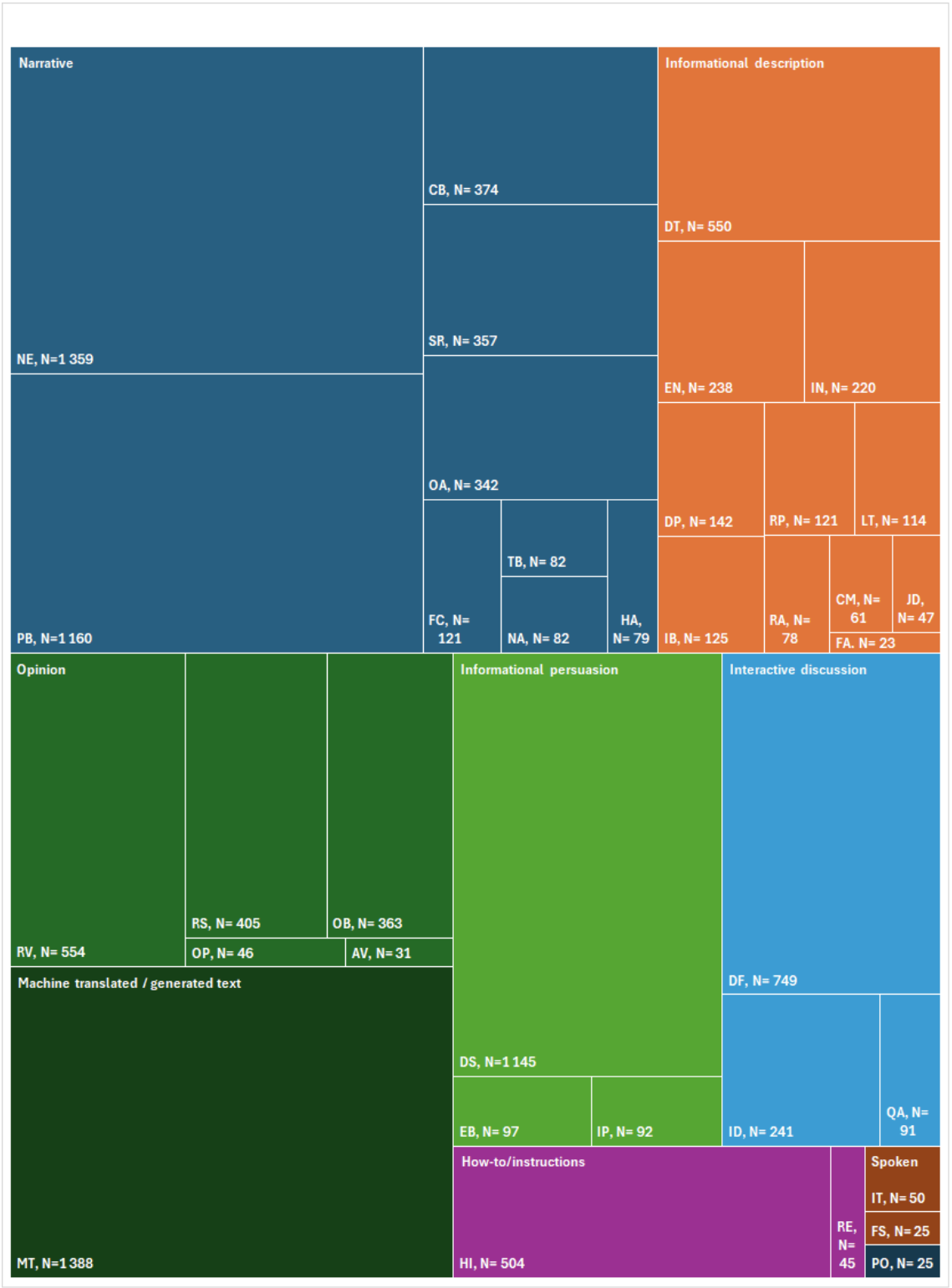

Figure 3.1: FinCORE register distribution

	N	F1-score
Narrative	754	0.86
Opinion	284	0.78
Informational description	362	0.72
Informational persuasion	252	0.77
Interactive discussion	231	0.84
How-to/instrcutions	117	0.71
Lyrical	5	0.13
Machine-translated/generated	276	0.98
Spoken	18	0.64

Table 3.1: FinCORE main register test set classification results

assigned register, which makes it a hybrid. There are 810 hybrids total in the corpus, of which 581 in the training set.

The largest main register in the corpus is "Narrative" with 3 956 texts comprising 34.32 % of the whole corpus. On the other hand, the smallest main register, "Lyrical", only has 25 texts labelled to it. All examples that are tagged with a subregister have the label for the corresponding main register as well. A visualisation of the label distribution can be viewed in Figure 3.1 as well as in Appendix A.

Versions of the dataset have been used in several benchmarks [19], [44], [66], the most recent being Skantsi & Laippala [9], where they achieved F1-scores of 0.78 with monolingual FinBERT and 0.79 with multilingual XLM-RoBERTa (XLM-R) when using all the register labels, including hybrid examples. They also reported a CNN baseline of 0.6. In [44], Laippala et al. report an AUC score of 83.8, when training only with six main register labels: Narrative, Opinion, Informational description, Interactive discussion, Informational persuasion, and How-to.

Skantsi and Laippala [9] observe some variation in register-specific classification results. Machine-translated texts achieved the highest F-score 0.98 as a category, followed by Narrative, the largest main register, with F-score 0.86. The second largest main register, Informational Description (n=1 719, 14.91% of the full corpus), gets a considerably lower F-score of 0.72, but some main registers get high scores even with fewer examples, such as Interactive Discussion with F-score of 0.84 (n=1 081, 9.38% of the full corpus).

The authors note that many registers with a high number of examples are pre-

dicted well, such as Personal blogs, News reports and Descriptions with intent to sell. Examples of high-performing subregisters with low number of examples are sports reports, Question-answer forums, Research articles, Job descriptions, Reviews, Discussion Forums and Religious texts. All of these examples of structurally consistent registers, which might explain good performance even with proportionally low coverage [9].

Respectively, registers such as Reports, FAQs, Poems, Course materials, Advice and Informational blogs with few examples don't perform too well. On the other hand, some registers have many examples and still underperform, e.g. Magazine/online articles and Community blogs. This could be explained by the lack of clear structure or characteristics and inner variation within the register [10].

3.1.2 Finnish Jigsaw Toxicity Challenge Dataset

As a second multilabel task, I am using TurkuNLP's machine translated version of Jigsaw Toxicity dataset [2], specifically the version translated from English to Finnish with DeepL. The corpus consists of 223 549 comments on Wikipedia talk page discussions and is labelled in six categories: "Toxicity", "Severe toxicity", "Threat", "Obscene", "Insult", and "Identity attack". The majority of the comments have not been assigned any of these labels, and comments without any labels are treated as a seventh, "Clean", category [12]. The full label distribution can be viewed in Table 3.2.

	Train	Test	Total
Identity attack	1 405	712	2 117 (0.95 %)
Insult	7 877	3 427	11 304 (5.05 %)
Obscene	8 449	3 691	12 140 (5.43 %)
Severe toxicity	1 595	367	1 962 (0.88 %)
Threat	478	211	689 (0.31 %)
Toxicity	15 924	6 090	22 014 (9.85 %)
Clean	143 346	57 735	201 081 (89.95 %)

Table 3.2: Toxicity dataset label distribution

[2]https://huggingface.co/datasets/TurkuNLP/jigsaw_toxicity_pred_fi

Eskelinen et al. [12] report an F-score of 0.66 for FinBERT and 0.65 for XLM-RoBERTa, which are comparable to the original 0.69 F-score of the fine-tuned English BERT model. From now on, I will refer to the FinBERT's higher F-score as a benchmark for this task.

The authors highlight that the task's ambiguity might influence the classification results. Even after weighing the labels to favour the smaller classes, they observed a large number of misclassifications into the "Clean" category, which accounts for over 89 % of the dataset and is the largest of the seven categories. This might also be due to the difficulty of the task or the subjectivity involved in annotation. Additionally, they suggest that subtle differences in nuance introduced by machine translation may also contribute to these errors. Another significant observation in their study is the frequent misclassification of examples from the "Severe toxicity" and "Threat" categories, which were under-predicted.

3.1.3 Yle Corpus

The Yle corpus [65] has previously been used to evaluate the Finnish BERT model, FinBERT [2]. The dataset is created with Sampo Pyysalo's tools [3], and contains 120 000 news articles, each labelled with one of the 10 most frequent topics, 12 000 examples per label. Virtanen et al. [2] report a 91.76% accuracy on a text classification task with FinBERT uncased, fine-tuned with a balanced dataset of 100 000 examples.

3.1.4 Ylilauta Corpus

Similar to the Yle corpus, the Ylilauta corpus [64] used in FinBERT evaluation [2] is labelled into 10 classes, one class per example and is created with Sampo Pyysalo's

[3]https://github.com/spyysalo/yle-corpus

tools [4]. In contrast to the Yle corpus' formal news texts, the Ylilauta consists of 120 000 examples from a Finnish online discussion forum, where Virtanen et al. [2] have chosen the most frequent categories, 12 000 examples per label. They report an 82.20% accuracy when fine-tuning FinBERT uncased with 100 000 examples balanced by class.

The authors note that this corpus performs considerably better with the monolingual FinBERT compared to the multilingual BERT, which might stem from the data used in the models' training: whereas FinBERT training material contains informal Finnish, their comparison models, fastText embedding models and multilingual BERT, do not.

3.2 Experimental Setup

The results from Tunstall et al. [13] are promising in several aspects. First, they offer a solution for limited amount of data, thus alleviating the cost for annotations. Secondly, their method uses a fraction of the resources fine-tuning a traditional Transformer model does, making the training process more accessible to lower-end devices. Lastly, they present performance close to that of the fine-tuned Transformer models with several different classification tasks. In this thesis, I will try to replicate their success and test out Finnish classification tasks with artificially restricted datasets.

For the choice of different sentence embedding models to be fine-tuned, I have used the monolingual Finnish paraphrase model [5] [31] as a baseline. As preliminary criteria, I have used the following:

1. The model must support Finnish language.

[4]https://github.com/spyysalo/ylilauta-corpus

[5]https://huggingface.co/TurkuNLP/sbert-cased-finnish-paraphrase

2. The model must be around 1GB or preferably smaller.

To guide my choice in the sea of embedding models, I used Huggingface's Massive Text Embedding Benchmark leaderboard [6] for English tasks to choose three models: multilingual-e5-small [7], paraphrase-multilingual-MiniLM-L12-v2 [8], and paraphrase-multilingual-mpnet-base-v2 [9], which has also been used in the multilingual section of the original paper. Since the leaderboard does not explicitly include Finnish language, this might have left out some potential models. In addition, I decided to test paraphrase-xlm-r-multilingual-v1 [10], since it is based on the cross-lingual XLM-RoBERTa that has fared well in the benchmarks. In the end, I chose to compare five different models, which are described in more detail in Table 3.3. For clarity, from now on I will refer to the models with the aliases listed in the table.

Model	Alias	Parameters	Size	MTEB
sbert-cased-finnish-paraphrase	FinSBERT	125M	0.50GB	-
multilingual-e5-small	e5-small	118M	0.44GB	124
paraphrase-multilingual-MiniLM-L12-v2	MiniLM	118M	0.44GB	148
paraphrase-multilingual-mpnet-base-v2	MPNet	278M	1.04GB	143
paraphrase-xlm-r-multilingual-v1	XLM-R	278M	1.11GB	-

Table 3.3: Sentence embedding models chosen for the experiments

As per Tunstall et al. [13], I will compose my experiments by fine-tuning the embedding models for 1 epoch with stable learning rate with no further parameter optimisation. For my experiments, I have chosen to use learning rate of 2e-5, which is the default learning rate provided by the SetFit library. To compare performance across different training data set sizes, I will divide the training set for each task into sets of 8, 16, 32 and 64 examples per class label. In multilabel tasks, each label might have fewer or more matches, depending on the label distribution in the dataset. In order to take variation into account, each sample size will have 10

[6] https://huggingface.co/spaces/mteb/leaderboard

[7] https://huggingface.co/intfloat/multilingual-e5-small

[8] https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

[9] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

[10] https://huggingface.co/sentence-transformers/paraphrase-xlm-r-multilingual-v1

randomised training sets that will be used when fine-tuning the different embedding models. In the fine-tuning step, I will use the validation sets provided with the datasets, or in the case of the toxicity challenge dataset, I will split the training set with the same set sizes as Eskelinen et al. [12] before further division into the different training sets. This might not be necessary, since I'm only going to train each model for one epoch, and I'm not going to carry out any parameter optimisation or, for example, use early stopping to prevent overfitting. The final results of the experiment will be reported based on the models' predictions on the provided test datasets. Something to note is that several classes in the FinCORE corpus have fewer than 64 or 32 examples: this means that it is impossible to create unique, balanced training sets with as many examples.

As mentioned in 2.3.1, the SetFit training process consists of two steps: fine-tuning the sentence embedding model and training a classification head on top of the fine-tuned model. As a baseline, I will use the results of exposing the smallest datasets to the classification head only, without fine-tuning the sentence embedding models, as has been done in previous studies with using Sentence Transformers for text classification [54], [55]. I will then compare the results from fine-tuning with different size datasets to the original benchmarks as well as the baseline results to evaluate the models' performance. To validate my implementation, I have tested the emotion recognition task from [13] and received results comparable to the ones in the original paper.

All the experiments have been run on Finnish IT Center for Science Puhti supercomputer [11]. The full code implementation of the experiments is available on GitHub [12].

[11] https://csc.fi/en/

[12] https://github.com/annsaln/setfit-classifier

3.2.1 Evaluation

	Positive	Negative
Predicted positive	True positive (TP)	False positive (FP)
Predicted negative	False negative (TN)	False Negative (FN)

Table 3.4: Confusion matrix

Evaluating text classification tasks can be conducted by using several different metrics. One of the simplest ways to evaluate a model's performance is to use confusion matrix to establish the model's precision, recall and accuracy [22]. Table 3.4 demonstrates the composition of the values used in the calculation, and the formulas for each of these metrics are the following:

$$precision = \frac{TP}{TP + FP} \tag{3.1}$$

$$recall = \frac{TP}{TP + FN} \tag{3.2}$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.3}$$

Simply put, accuracy measures the percentage of examples the model has labelled correctly. Accuracy can work well in situations, where the dataset is balanced. Such is the case of Yle and Ylilauta corpora, for which accuracy has been used as an evaluation metric [2]. In cases, where the data is imbalanced, like in the Toxicity challenge dataset, accuracy might seem high even when only predicting one class that comprises nearly 90% of the dataset.

Because of the nature of the accuracy metric, it is rarely used in text classification evaluation [22]. Instead, it is useful to take a look at a model's precision (P) and recall (R) metrics. A common way to measure a model's performance is to use both to calculate their balanced representation, an F-score [22]:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{3.4}$$

The β parameter denotes the weight one desires to put on the precision and recall of the model. Values below 1 will favour precision, whereas values above 1 will favour recall. The simplest way is to use $\beta = 1$ to create a F1-score with equal weighing on both precision and recall:

$$F_1 = \frac{2PR}{P + R} \tag{3.5}$$

In this research, I will be mainly using F1-score as an evaluation metric to validate each model. Some of the tasks I've chosen do not have a benchmark that uses this metric, but I've chosen to use it for all experiments for ease of comparison in the scope of this work.

4 Results

In this chapter, I will present the results that were obtained in the experiments described in Chapter 3. I will compare the sentence embedding models' performance across different classification tasks and with different size training sets. To establish a baseline for each task and model, I've chosen to use the SetFit library to fine-tune only the classification head with the smallest size datasets.

The different models are evaluated by inspecting the mean of micro average F1-score of 10 unique training instances, as well as the F1-score standard deviation (SD) and a mean difference to the baseline (BL). This is done to take data variation into account, even though the models themselves are trained with equal amount of exposure to different labels whenever it is allowed by the training data label distribution. FinCORE dataset especially has classes with so few examples that it is impossible to create a training set where there are 64 or even 32 examples of certain labels. The training sets are referred to as the number of examples per label, and the 10 training sets per sample size remain the same across all different model experiments.

The tasks include two multilabel datasets, FinCORE and the Jigsaw toxicity challenge dataset. The evaluation of the multilabel models by assigning labels that receive a sigmoid output result that exceeds or equals the threshold of 0.5. The Yle and Ylilauta tasks use the maximum value of the softmax output to assign a predicted label to an example.

4.1 FinCORE

I have chosen to split the FinCORE task into two parts: a. full corpus with all the subregister and main register labels and b. a more restricted approach with main register labels only. This is done due to the complexity of the corpus, and also to compare the method's performance with the same dataset in varying granularity. Both of these approaches are true multilabel tasks, meaning that one example might have several true labels assigned to it. The averaged model results can be found in tables 4.1 and 4.2.

	baseline	n = 8	n = 16	n = 32	n = 64
FinSBERT	F1 0.34	F1 0.46	F1 0.50	F1 0.51	F1 **0.53**
	SD 0.01	SD 0.02	SD 0.01	SD 0.01	SD 0.2
MiniLM	F1 0.27	F1 0.33	F1 0.38	F1 0.40	F1 **0.43**
	SD 0.01	SD 0.02	SD 0.01	SD 0.01	SD 0.01
e5-small	F1 0.0002	F1 0.39	F1 0.46	F1 0.50	F1 **0.52**
	SD 0.0002	SD 0.02	SD 0.02	SD 0.01	SD 0.01
XLM-R	F1 0.30	F1 0.38	F1 0.39	F1 0.39	F1 **0.42**
	SD 0.01	SD 0.01	SD 0.01	SD 0.01	SD 0.01
MPNet	F1 0.23	F1 0.35	F1 0.37	F1 0.39	F1 **0.43**
	SD 0.02	SD 0.02	SD 0.02	SD 0.02	SD 0.02

Table 4.1: FinCORE classification results for different sample sizes

	baseline	n = 8	n = 16	n = 32	n = 64
FinSBERT	F1 0.24	F1 0.42	F1 0.58	F1 0.64	F1 **0.68**
	SD 0.04	SD 0.04	SD 0.02	SD 0.01	SD 0.01
MiniLM	F1 0.12	F1 0.13	F1 0.32	F1 0.48	F1 **0.54**
	SD 0.04	SD 0.05	SD 0.04	SD 0.02	SD 0.01
e5-small	F1 0.0	F1 0.04	F1 0.47	F1 0.63	F1 **0.68**
	SD 0.0	SD 0.04	SD 0.06	SD 0.02	SD 0.01
XLM-R	F1 0.17	F1 0.24	F1 0.46	F1 0.52	F1 **0.56**
	SD 0.03	SD 0.05	SD 0.03	SD 0.01	SD 0.01
MPNet	F1 0.07	F1 0.19	F1 0.43	F1 0.50	F1 **0.55**
	SD 0.03	SD 0.06	SD 0.02	SD 0.02	SD 0.02

Table 4.2: FinCORE main register classification results for different sample sizes

In both tasks, TurkuNLP's monolingual FinSBERT model reaches the best F1-scores when evaluated with the test set. In the main registers only task, the multilingual e5-small is tied for the first place, whereas in the full label set task, it comes a close second. What makes the two models' performance differ is their learning curve: FinSBERT reaches the highest baseline results, when e5-small's baseline is practically zero. This might indicate that e5-small greatly benefits from task-specific

fine-tuning, even with just a few examples.

For some reason, the task with all labels included gets more stable results with fewer examples. This being a multilabel task, it might be that some classes get more exposure than others. It might also be a result of intraclass variance, as observed in [10]. In the end, with the largest sample size datasets, all models reach higher scores in the main register task. No model, however, comes too close to the original benchmark, F1-score 0.69 with the full label set.

	FinSBERT n = 64	MiniLM n = 64	e5-small n = 64	XLM-R n = 64	MPNet n = 64
Narrative (n=754)	F1 **0.74** SD 0.02 BL 0.19 Δ 0.56	F1 0.64 SD 0.1 BL 0.07 Δ 0.57	F1 0.73 SD 0.01 BL 0.0 Δ 0.73	F1 0.65 SD 0.01 BL 0.12 Δ 0.53	F1 0.62 SD 0.04 BL 0.01 Δ 0.61
Opinion (n=284)	F1 0.60 SD 0.02 BL 0.30 Δ 0.30	F1 0.55 SD 0.02 BL 0.19 Δ 0.35	F1 **0.65** SD 0.03 BL 0.0 Δ 0.65	F1 0.54 SD 0.02 BL 0.26 Δ 0.28	F1 0.55 SD 0.02 BL 0.14 Δ 0.41
Informational description (n=362)	F1 **0.59** SD 0.03 BL 0.14 Δ 0.44	F1 0.46 SD 0.03 BL 0.05 Δ 0.41	F1 0.58 SD 0.03 BL 0.0 Δ 0.58	F1 0.47 SD 0.01 BL 0.09 Δ 0.38	F1 0.49 SD 0.04 BL 0.01 Δ 0.48
Informational persuasion (n=252)	F1 0.62 SD 0.02 BL 0.19 Δ 0.44	F1 0.46 SD 0.02 BL 0.11 Δ 0.36	F1 **0.64** SD 0.03 BL 0.0 Δ 0.64	F1 0.49 SD 0.02 BL 0.13 Δ 0.36	F1 0.48 SD 0.03 BL 0.04 Δ 0.48
Interactive discussion (n=231)	F1 0.71 SD 0.02 BL 0.14 Δ 0.57	F1 0.53 SD 0.03 BL 0.02 Δ 0.52	F1 **0.71** SD 0.02 BL 0.0 Δ 0.71	F1 0.58 SD 0.04 BL 0.09 Δ 0.49	F1 0.59 SD 0.03 BL 0.003 Δ 0.58
How-to / instructions (n=117)	F1 0.51 SD 0.03 BL 0.19 Δ 0.33	F1 0.43 SD 0.02 BL 0.08 Δ 0.35	F1 **0.57** SD 0.03 BL 0.0 Δ 0.57	F1 0.44 SD 0.02 BL 0.20 Δ 0.23	F1 0.44 SD 0.02 BL 0.03 Δ 0.42
Lyrical (n=5)	F1 0.24 SD 0.07 BL 0.18 Δ 0.07	F1 0.11 SD 0.05 BL 0.10 Δ 0.01	F1 **0.37** SD 0.13 BL 0.0 Δ 0.37	F1 0.12 SD 0.05 BL 0.08 Δ 0.04	F1 0.15 SD 0.06 BL 0.12 Δ 0.03
Machine translated/generated (n=276)	F1 **0.95** SD 0.01 BL 0.59 Δ 0.36	F1 0.68 SD 0.05 BL 0.33 Δ 0.35	F1 0.91 SD 0.01 BL 0.0 Δ 0.91	F1 0.71 SD 0.03 BL 0.40 Δ 0.31	F1 0.68 SD 0.04 BL 0.29 Δ 0.38
Spoken (n=18)	F1 **0.24** SD 0.04 BL 0.05 Δ 0.19	F1 0.10 SD 0.10 BL 0.01 Δ 0.10	F1 0.21 SD 0.03 BL 0.0 Δ 0.21	F1 0.11 SD 0.03 BL 0.06 Δ 0.05	F1 0.13 SD 0.03 BL 0.0 Δ 0.13

Table 4.3: FinCORE main registers label specific average results per class and best model sample

The label-specific evaluation brings us some additional insight into the model comparison, as seen in Table 4.3. None of the models can perform as well as the benchmark, except in the "Lyrical" class, where three of the models outperform the

benchmark (F1 = 0.13 [9]). However, the standard deviation for this class is high, ranging from 0.05 to 0.13, and a similarly sized class "Spoken" gets significantly worse results compared to the benchmark (F1 = 0.64 [9]) with all the models.

While FinSBERT and e5-small perform in very similar ways at their best, there are some slight differences on the register level when comparing the top-performing instances. FinSBERT seems to do slightly better in registers "Narrative", "Informational description", "Machine translated / generated" and "Spoken", whereas e5-small outperforms it in registers "Opinion", "Informational persuasion", "How-to / instructions", and "Lyrical". The largest differences are in the classification results of "Lyrical", "How-to / instructions", and "Opinion" registers, all in favour of e5-small.

The greatest difference to the baseline when training with the FinSBERT model is observed with classes "Narrative", "Interactive discussion" and "Informational description". From these three, "Narrative" and "Informational description" are classes where FinSBERT also outperforms the rest of the models. E5-small overall has the largest difference to baseline, which is 0.0 for all the classes.

4.2 Toxicity Challenge Dataset

Like the FinCORE dataset, the Toxicity challenge dataset is also a multilabel dataset. One exception to the rule is that the "Clean" label does not co-occur with the rest of the labels but instead indicates a lack of any of the other labels. Other speciality of this dataset is that it has been machine translated from English to Finnish. The classification results for this task can be viewed in Table 4.4.

The results are low compared to the original benchmark (F1-score = 0.66 [12]), and the standard deviation is the highest of all the tasks across all models. This might be an indicator of the difficulty of the task. The model that performs the best on average is FinSBERT with 64 examples per label with mean F1-score of 0.53

	baseline	n = 8	n = 16	n = 32	n = 64
FinSBERT	F1 0.26	F1 0.36	F1 0.35	F1 0.44	F1 **0.53**
	SD 0.04	SD 0.09	SD 0.06	SD 0.05	SD 0.06
MiniLM	F1 0.21	F1 0.43	F1 **0.49**	F1 0.48	F1 0.40
	SD 0.03	SD 0.10	SD 0.09	SD 0.06	SD 0.16
e5-small	F1 0.10	F1 0.13	F1 0.25	F1 **0.37**	F1 0.36
	SD 0.0	SD 0.02	SD 0.05	SD 0.03	SD 0.14
XLM-R	F1 0.26	F1 0.40	F1 0.39	F1 0.40	F1 **0.44**
	SD 0.06	SD 0.13	SD 0.06	SD 0.11	SD 0.17
MPNet	F1 0.20	F1 0.35	F1 0.39	F1 0.42	F1 **0.43**
	SD 0.03	SD 0.13	SD 0.10	SD 0.07	SD 0.17

Table 4.4: Toxicity classification results for different sample sizes (F1-score, standard deviation in parenthesis)

(SD = 0.06). Out of these instances, the best model got an F1-score of 0.62, which is not too far from the benchmark. The best model overall is, surprisingly, a clear outlier of a MiniLM model trained with only 16 examples per label with an F1-score of 0.67. Something to note is that all models experience some kind of relapse while increasing the training set size: even if the best performance is achieved with the largest training set, there is some regression when switching from a smaller dataset to a larger one in all models except MPNet.

Sample size	FinSBERT n = 64	MiniLM n = 16	e5-small n = 32	XLM-R n = 64	MPNet n = 64
Identity attack (n=712)	F1 0.016	F1 **0.03**	F1 0.0	F1 0.01	F1 0.0
	SD 0.03	SD 0.03	SD 0.0	SD 0.03	SD 0.01
	BL 0.13	BL 0.16	BL 0.0	BL 0.16	BL 0.22
	Δ -0.12	Δ -0.14	Δ 0.0	Δ -0.15	Δ -0.22
Insult (n=3 427)	F1 0.31	F1 **0.32**	F1 0.18	F1 0.23	F1 0.22
	SD 0.06	SD 0.05	SD 0.01	SD 0.07	SD 0.06
	BL 0.19	BL 0.19	BL 0.10	BL 0.19	BL 0.20
	Δ 0.13	Δ 0.14	Δ 0.08	Δ 0.04	Δ 0.01
Obscene (n=3 691)	F1 0.32	F1 **0.36**	F1 0.19	F1 0.24	F1 0.22
	SD 0.06	SD 0.06	SD 0.01	SD 0.07	SD 0.06
	BL 0.18	BL 0.18	BL 0.11	BL 0.18	BL 0.19
	Δ 0.14	Δ 0.19	Δ 0.08	Δ 0.06	Δ 0.04
Severe toxicity (n=367)	F1 0.08	F1 **0.18**	F1 0.0	F1 0.0	F1 0.0
	SD 0.06	SD 0.08	SD 0.0	SD 0.0	SD 0.0
	BL 0.06	BL 0.09	BL 0.0	BL 0.09	BL 0.12
	Δ 0.02	Δ 0.10	Δ 0.0	Δ -0.09	Δ -0.12
Threat (n=211)	F1 **0.04**	F1 0.04	F1 0.0	F1 0.0	F1 0.0
	SD 0.03	SD 0.05	SD 0.0	SD 0.0	SD 0.0
	BL 0.12	BL 0.14	BL 0.0	BL 0.14	BL 0.21
	Δ -0.08	Δ -0.10	Δ 0.0	Δ -0.14	Δ -0.21
Toxicity (n=6 090)	F1 0.38	F1 **0.38**	F1 0.27	F1 0.33	F1 0.32
	SD 0.06	SD 0.04	SD 0.01	SD 0.08	SD 0.08
	BL 0.23	BL 0.22	BL 0.17	BL 0.22	BL 0.22
	Δ 0.15	Δ 0.16	Δ 0.10	Δ 0.10	Δ 0.10
Clean (n=57 735)	F1 **0.71**	F1 0.59	F1 0.59	F1 0.60	F1 0.59
	SD 0.05	SD 0.10	SD 0.04	SD 0.30	SD 0.29
	BL 0.38	BL 0.24	BL 0.0	BL 0.24	BL 0.19
	Δ 0.33	Δ 0.36	Δ 0.59	Δ 0.36	Δ 0.39

Table 4.5: Toxicity label level results from the best performing batches

As we see in Table 4.5, performance is not consistent through all the categories: with most models, "Identity attack", "Severe toxicity" and "Threat" classes are severely under predicted. In fact, there is some indication of catastrophic forgetting, since the baseline results in these classes are in many cases higher than with the best performing models. Such is the case of "Identity attack" with FinSBERT, MiniLM, XLM-R and MPNet models, "Severe toxicity" with XLM-R and MPNet models as well as "Threat" with FinSBERT, MiniLM, XLM-R and MPNet models. Since e5-small baselines are already 0.0 in these classes, there isn't any unlearning to do. This might also be related to the regression experienced when training with larger datasets.

The largest difference within category is in the prediction results for "Severe toxicity", where MiniLM reaches F1-score 0.18, FinSBERT 0.08, and the rest of the models regressing to or staying at F1-score 0.0. "Clean" is another category with a clear best performer: FinSBERT gets roughly 0.10 points higher F1-scores than the other models. The greatest positive difference to the baseline can be observed in the "Clean" category with all models, where they also reach the highest label-level F1-scores.

As the highest F1-scores go consistently to the clean category, one could argue that this method succeeds mainly in differentiating clean texts from toxic ones. This might be due to SetFit's training procedure: since SetFit uses contrastive learning for fine-tuning the embedding model, it will create distance between examples that do belong to different classes. Because the "Clean" label does not coexist with any other labels, it should be easier to single out in the vector space. Figure 4.1 illustrates the evolution of FinSBERT models fine-tuned with increasing data: while the baseline model's embeddings show all the classes jumbled in one mass, as the models are fine-tuned with more data, the blue "Clean" category starts to separate from the rest of the classes. However, one can observe that the distinction between

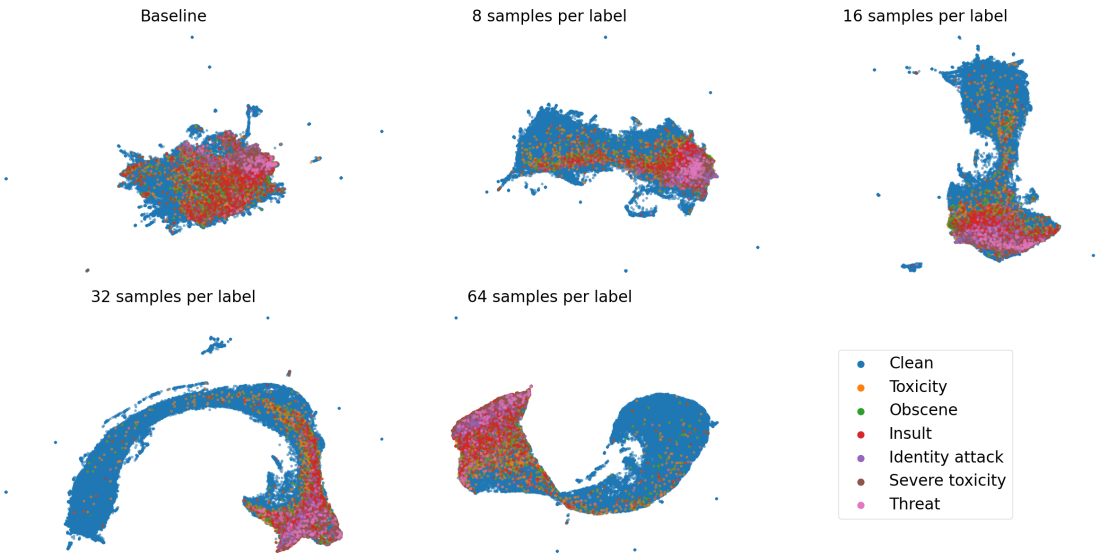

Figure 4.1: FinSBERT embeddings fine-tuned with Toxicity challenge data

"Clean" examples and the rest is not perfect even when fine-tuning with the largest sample size. The data used for creating these embeddings is incremental, meaning that all the training data points from the smaller datasets are included in the larger datasets as well.

Label combination	Predictions
Clean	35 584
Insult, Obscene, Toxicity	25 648
Toxicity	1 791
-	846
Insult, Toxicity	109

Table 4.6: Test set predictions from the best instance of e5-small

A notable difference to the other tasks here is the poor performance of e5-small. While with FinCORE and Yle datasets it performs on par with the best model, here it gets the worst results of all, and seems to have trouble learning. The baseline prediction F1-score for four out of seven classes is 0.0, and even after fine-tuning, the results leave much to be desired. With the best instance of e5-small, "Identity attack", "Severe toxicity" and "Threat" classes are never once predicted when evaluating the test set, and the predictions are basically split between "Clean" and

"Insult"+"Obscene"+"Toxicity" combination, as shown in Table 4.6. When compared to the true test set label distribution shown in Table 3.2, it is clear that while the Clean category remains the largest predicted class, it is under-predicted, and the labels in the second largest prediction category are severely over-predicted.

All tested models seem to prefer these two topmost label combinations shown in Table 4.6 at the expense of others, even though not as harshly as e5-small. With MiniLM, the classes are more distributed, but it looks like as the dataset size grows, the model will start preferring the toxic label categories: with a 64 example iteration, a model that had previously predicted most of the examples (correctly) to the "Clean" class starts to prefer the same label combination as above, and only 5 504 examples go to the "Clean" category.

4.3 Yle Corpus

Contrary to the two previous tasks, the Yle task is a multiclass classification task, rather than a multilabel one. Each example receives only one label out of the ten balanced classes. The results of the experiments with this dataset can be viewed in Table 4.7. .

	baseline	n = 8	n = 16	n = 32	n = 64
FinSBERT	F1 0.58	F1 0.75	F1 0.83	F1 0.85	F1 **0.86**
	SD 0.03	SD 0.02	SD 0.01	SD 0.01	SD 0.004
MiniLM	F1 0.73	F1 0.78	F1 0.80	F1 0.81	F1 **0.83**
	SD 0.02	SD 0.02	SD 0.01	SD 0.01	SD 0.004
e5-small	F1 0.75	F1 0.81	F1 0.84	F1 0.85	F1 **0.85**
	SD 0.01	SD 0.02	SD 0.01	SD 0.01	SD 0.01
XLM-R	F1 0.58	F1 0.79	F1 0.81	F1 **0.83**	F1 0.82
	SD 0.02	SD 0.01	SD 0.01	SD 0.01	SD 0.01
MPNet	F1 0.71	F1 0.80	F1 0.82	F1 **0.83**	F1 0.82
	SD 0.02	SD 0.02	SD 0.01	SD 0.01	SD 0.01

Table 4.7: Yle classification results for different sample sizes

The performance across different models is surprisingly similar and stable with this task. While FinSBERT reaches the best overall F1-score of 0.86, the rest are not far behind. The standard deviation is also very low, especially compared to the

Toxicity task results. The previously reported benchmark of 0.92 accuracy [2] is not far away, and here it would seem that the SetFit method succeeds fairly well even with a smaller amount of data. This is supported by the fact that the XLM-R and MPNet performance plateaus with 32 examples per label, and the other models gain only slight improvements to their results when doubling the dataset size to 64.

However, it should be noted that the baseline F1-scores are also quite high, and reasonably good results are achievable even without fine-tuning the sentence embedding models. The FinSBERT and XLM-R models benefit the most from fine-tuning as their baseline results are lower, but the other models gain only a moderate increase in their performance. This contradicts the results from the FinCORE and Toxicity tasks, where these two models have reached considerably higher baseline results than the rest.

The class-specific results seen in Table 4.8 reflect the results in Table 4.7. The performance is relatively stable across all classes with all models, with only minor differences. FinSBERT outperforms the other models slightly in all categories but one, the greatest difference being in the class "Talous"[1]. FinSBERT and e5 get nearly identical F1-scores in categories "Koulutus ja kasvatus"[2], "Liikenne ja kuljetus"[3], "Onnettomuudet"[4], "Terveys"[5], and "Urheilu"[6].

The highest F1-scores go to "Urheilu" with all models, its F1-scores ranging from 0.94 to 0.97. The class's baseline results are also rather high, even without fine-tuning the sentence embedding model. These observations are on par with the remarks in [9], where it is noted that sports reports are predicted with high accuracy even when the model is trained with a small number of examples. According to the

[1]Economy (author's translation)

[2]Education and upbringing (author's translation)

[3]Traffic and transport (author's translation)

[4]Accidents (author's translation)

[5]Health (author's translation)

[6]Sports (author's translation)

Sample size	FinBERT n = 64	MiniLM n = 64	e5-small n = 64	XLM-R n = 32	MPNet n = 32
Koulutus ja kasvatus (n=1 000)	F1 **0.90**	F1 0.89	F1 0.90	F1 0.88	F1 0.89
	SD 0.005	SD 0.005	SD 0.01	SD 0.01	SD 0.01
	BL 0.70	BL 0.84	BL 0.81	BL 0.70	BL 0.78
	Δ 0.20	Δ 0.04	Δ 0.09	Δ 0.18	Δ 0.11
Liikenne ja kuljetus (n=1 000)	F1 **0.84**	F1 0.81	F1 0.84	F1 0.81	F1 0.82
	SD 0.13	SD 0.01	SD 0.01	SD 0.01	SD 0.01
	BL 0.61	BL 0.71	BL 0.68	BL 0.56	BL 0.69
	Δ 0.23	Δ 0.09	Δ 0.16	Δ 0.25	Δ 0.13
Kulttuuri (n=1 000)	F1 **0.87**	F1 0.82	F1 0.86	F1 0.82	F1 0.83
	SD 0.01	SD 0.01	SD 0.01	SD 0.02	SD 0.01
	BL 0.49	BL 0.67	BL 0.72	BL 0.46	BL 0.58
	Δ 0.38	Δ 0.15	Δ 0.14	Δ 0.36	Δ 0.25
Luonto (n=1 000)	F1 **0.86**	F1 0.79	F1 0.84	F1 0.80	F1 0.80
	SD 0.01	SD 0.0	SD 0.01	SD 0.01	SD 0.01
	BL 0.55	BL 0.70	BL 0.73	BL 0.55	BL 0.69
	Δ 0.31	Δ 0.09	Δ 0.11	Δ 0.25	Δ 0.11
Onnettomuudet (n=1 000)	F1 0.86	F1 0.84	F1 **0.86**	F1 0.82	F1 0.83
	SD 0.02	SD 0.02	SD 0.01	SD 0.03	SD 0.03
	BL 0.63	BL 0.70	BL 0.74	BL 0.61	BL 0.73
	Δ 0.24	Δ 0.15	Δ 0.12	Δ 0.21	Δ 0.10
Politiikka (n=1 000)	F1 **0.82**	F1 0.78	F1 0.80	F1 0.78	F1 0.78
	SD 0.02	SD 0.02	SD 0.02	SD 0.02	SD 0.02
	BL 0.55	BL 0.70	BL 0.72	BL 0.56	BL 0.71
	Δ 0.27	Δ 0.08	Δ 0.08	Δ 0.22	Δ 0.07
Rikokset (n=1 000)	F1 **0.88**	F1 0.84	F1 0.86	F1 0.85	F1 0.85
	SD 0.01	SD 0.01	SD 0.01	SD 0.01	SD 0.01
	BL 0.62	BL 0.75	BL 0.74	BL 0.63	BL 0.76
	Δ 0.25	Δ 0.09	Δ 0.12	Δ 0.21	Δ 0.09
Talous (n=1 000)	F1 **0.79**	F1 0.71	F1 0.76	F1 0.74	F1 0.73
	SD 0.01	SD 0.02	SD 0.02	SD 0.01	SD 0.02
	BL 0.45	BL 0.56	BL 0.65	BL 0.43	BL 0.58
	Δ 0.34	Δ 0.13	Δ 0.11	Δ 0.31	Δ 0.15
Terveys (n=1 000)	F1 **0.86**	F1 0.83	F1 0.86	F1 0.83	F1 0.83
	SD 0.01	SD 0.01	SD 0.01	SD 0.01	SD 0.02
	BL 0.54	BL 0.74	BL 0.77	BL 0.64	BL 0.75
	Δ 0.33	Δ 0.09	Δ 0.09	Δ 0.19	Δ 0.08
Urheilu (n=1 000)	F1 **0.97**	F1 0.94	F1 0.97	F1 0.95	F1 0.95
	SD 0.01	SD 0.01	SD 0.005	SD 0.01	SD 0.01
	BL 0.69	BL 0.86	BL 0.88	BL 0.68	BL 0.80
	Δ 0.28	Δ 0.08	Δ 0.08	Δ 0.27	Δ 0.15

Table 4.8: Yle classification label level results

authors, this might be due to the often formulaic structure of the texts. "Talous" receives the worst F1-scores with all the models, ranging from 0.73 to 0.79.

The class "Kulttuuri"[7] appears to have benefitted the most from fine-tuning, when comparing the final results to the baseline. With FinSBERT, the difference to baseline is 0.38, and the class ends up being among the best performing ones. Also, XLM-R and MPNet have "Kulttuuri" as the class where the most improvement can be seen. With MiniLM, "Kulttuuri" is accompanied by "Liikenne ja kuljetus", whereas e5-small has the largest difference to the baseline in the class "Onnettomuudet". The smallest improvement can be seen in "Koulutus ja kasvatus" with FinSBERT, MiniLM and XLM-R models, "Urheilu" with e5-small and "Politiikka" with MPNet.

4.4 Ylilauta Corpus

Similarly to the Yle task, the Ylilauta task consists of multiclass classification into ten balanced classes. The main difference here is the register: the Yle corpus contains news texts from ten different categories, whereas the Ylilauta corpus is gathered from a discussion forum and its ten most common discussion topic themes. Contrary to the Yle corpus, the Ylilauta corpus contains also informal language, and the texts might be more non-formulaic than news texts in general. The classification results for this task can be viewed in Table 4.9.

Based on the results in Table 4.9, it looks like FinSBERT outperforms the rest of the models by a large margin with an F1-score of 0.74. Otherwise, the rest of the models perform at a similar level with F1-scores around 0.60. While the previously reported accuracy of 0.82 with FinBERT [2] is not reached, the FinSBERT model does perform relatively well when considering the small number of examples compared to the original 10 000 per class. Even though most of the models reach their

[7]Culture (author's translation)

	baseline	n = 8	n = 16	n = 32	n = 64
FinSBERT	F1 0.41	F1 0.54	F1 0.70	F1 0.73	F1 **0.74**
	SD 0.02	SD 0.04	SD 0.01	SD 0.01	SD 0.004
MiniLM	F1 0.45	F1 0.47	F1 0.51	F1 0.54	F1 **0.57**
	SD 0.02	SD 0.02	SD 0.02	SD 0.01	SD 0.01
e5-small	F1 0.48	F1 0.50	F1 0.57	F1 0.59	F1 **0.61**
	SD 0.02	SD 0.03	SD 0.02	SD 0.01	SD 0.01
XLM-R	F1 0.44	F1 0.51	F1 0.58	F1 **0.60**	F1 0.60
	SD 0.02	SD 0.02	SD 0.01	SD 0.01	SD 0.01
MPNet	F1 0.52	F1 0.54	F1 0.59	F1 **0.61**	F1 0.60
	SD 0.02	SD 0.02	SD 0.01	SD 0.01	SD 0.01

Table 4.9: Ylilauta classification results for different sample sizes

best performance with the largest dataset, the improvement when sizing up from 32 examples per class to 64 examples per class isn't enormous even with models that do benefit from the larger dataset. XLM-R and MPNet reach their best scores with only 32 examples per class.

The standard deviation scores, while on average higher than with the Yle task, are quite low. All models receive similar baseline F1-scores, with a 0.11 difference between the best (MPNet) and worst (FinSBERT) performing model. The performance gain is not great with all the models: for example, MPNet gains only 0.09 points increase from the baseline score with fine-tuning. However, FinSBERT benefits from fine-tuning, with 0.33 difference to the baseline.

The class-level classification results can be viewed in Table 4.10. The best performing class overall is "Ajoneuvot"[8] with FinSBERT F1-score of 0.84. The other models succeeded the best in the class "Televisio"[9], with F1-scores ranging from 0.71 to 0.74. The class "Hikky"[10] proved to be the most difficult one to predict for all models except MiniLM, which performed the worst in the class "Sota"[11].

The greatest improvement to the baseline score can be observed in the class "Penkkiurheilu"[12] with FinSBERT gaining 0.43 increase to the baseline with fine-

[8]Vehicles (author's translation)

[9]Television (author's translation)

[10]Hikikomori (author's translation)

[11]War (author's translation)

[12]Spectator sports (author's translation)

Sample size	FinSBERT n = 64	MiniLM n = 64	e5-small n = 64	XLM-R n = 32	MPNet n = 32
Ajoneuvot (n=1 000)	F1 **0.84**	F1 0.65	F1 0.70	F1 0.67	F1 0.70
	SD 0.01	SD 0.02	SD 0.01	SD 0.02	SD 0.02
	BL 0.51	BL 0.56	BL 0.60	BL 0.58	BL 0.63
	Δ 0.34	Δ 0.09	Δ 0.10	Δ 0.09	Δ 0.07
Hikky (n=1 000)	F1 **0.65**	F1 0.49	F1 0.50	F1 0.47	F1 0.49
	SD 0.01	SD 0.01	SD 0.02	SD 0.02	SD 0.03
	BL 0.34	BL 0.39	BL 0.38	BL 0.32	BL 0.42
	Δ 0.30	Δ 0.10	Δ 0.12	Δ 0.15	Δ 0.07
Kuntosali (n=1 000)	F1 **0.73**	F1 0.54	F1 0.57	F1 0.59	F1 0.58
	SD 0.01	SD 0.02	SD 0.01	SD 0.02	SD 0.02
	BL 0.42	BL 0.38	BL 0.46	BL 0.40	BL 0.48
	Δ 0.31	Δ 0.16	Δ 0.10	Δ 0.19	Δ 0.11
Muoti (n=1 000)	F1 **0.73**	F1 0.57	F1 0.57	F1 0.60	F1 0.61
	SD 0.01	SD 0.01	SD 0.02	SD 0.01	SD 0.02
	BL 0.40	BL 0.43	BL 0.39	BL 0.47	BL 0.53
	Δ 0.32	Δ 0.14	Δ 0.17	Δ 0.13	Δ 0.09
Pelit (n=1 000)	F1 **0.73**	F1 0.56	F1 0.67	F1 0.63	F1 0.62
	SD 0.01	SD 0.01	SD 0.01	SD 0.02	SD 0.02
	BL 0.32	BL 0.39	BL 0.51	BL 0.38	BL 0.47
	Δ 0.40	Δ 0.17	Δ 0.17	Δ 0.25	Δ 0.14
Penkkiurheilu (n=1 000)	F1 **0.77**	F1 0.63	F1 0.68	F1 0.66	F1 0.69
	SD 0.02	SD 0.01	SD 0.01	SD 0.02	SD 0.02
	BL 0.34	BL 0.51	BL 0.57	BL 0.48	BL 0.57
	Δ 0.43	Δ 0.12	Δ 0.12	Δ 0.18	Δ 0.12
Politiikka (n=1 000)	F1 **0.75**	F1 0.65	F1 0.65	F1 0.65	F1 0.67
	SD 0.01	SD 0.01	SD 0.02	SD 0.04	SD 0.02
	BL 0.43	BL 0.53	BL 0.54	BL 0.48	BL 0.60
	Δ 0.32	Δ 0.12	Δ 0.11	Δ 0.17	Δ 0.07
Seksuaalisuus (n=1 000)	F1 **0.70**	F1 0.49	F1 0.53	F1 0.54	F1 0.54
	SD 0.02	SD 0.02	SD 0.02	SD 0.02	SD 0.02
	BL 0.44	BL 0.42	BL 0.44	BL 0.40	BL 0.50
	Δ 0.26	Δ 0.07	Δ 0.09	Δ 0.14	Δ 0.05
Sota (n=1 000)	F1 **0.72**	F1 0.48	F1 0.53	F1 0.53	F1 0.53
	SD 0.01	SD 0.03	SD 0.02	SD 0.03	SD 0.03
	BL 0.45	BL 0.35	BL 0.29	BL 0.33	BL 0.40
	Δ 0.27	Δ 0.13	Δ 0.24	Δ 0.20	Δ 0.13
Televisio (n=1 000)	F1 **0.80**	F1 0.71	F1 0.74	F1 0.72	F1 0.73
	SD 0.01	SD 0.01	SD 0.01	SD 0.02	SD 0.01
	BL 0.46	BL 0.58	BL 0.56	BL 0.55	BL 0.68
	Δ 0.34	Δ 0.12	Δ 0.18	Δ 0.17	Δ 0.06

Table 4.10: Ylilauta label specific classification results

tuning. The other models benefit the most from fine-tuning in the classes "Pelit"[13] and "Sota". On the other hand, all the models see the least improvement in the class "Seksuaalisuus"[14], except for XLM-R, which learns the least in "Ajoneuvot".

The largest in-category variance is in the classes "Sota" and "Seksuaalisuus", with F1-scores ranging from 0.48 to 0.72 in "Sota", and from 0.49 to 0.70 in "Seksuaalisuus". Respectively, the smallest variance can be observed within the classes "Televisio" and "Politiikka"[15], where the F1-scores from all models are within 0.10 range.

[13]Games (author's translation)

[14]Sexuality (author's translation)

[15]Politics (author's translation)

5 Discussion

The hypothesis to my first research question "How well do the fine-tuned Sentence Transformer models perform with Finnish tasks compared to the reported benchmarks? Can the SetFit method reach results comparable to the state-of-the-art?" was that Sentence Transformer models fine-tuned by SetFit would not reach state-of-the-art results. As evidenced in Chapter 4, the hypothesis came true. This supports the results from earlier research, indicating that SetFit might not be the best solution for fine-tuning a model when the highest possible accuracy is necessary. However, the results from this research show promise for the more data-sparse tasks: in the event that one might need a classification model for a clearly defined task with only a few existing examples, SetFit could be a solution worth testing. In this chapter, I will further discuss the possible benefits, drawbacks and limitations of the method I have encountered while doing this research, as well as considerations for anyone interested in testing SetFit themselves.

The second research question was interested in comparing the different Sentence Transformer models and if there would be a difference in the performance between multilingual and monolingual Finnish models when classifying Finnish text. In Chapter 4, it was shown that the monolingual FinSBERT was a stable performer and best in all tasks. The difference to the following models varied, and the least difference was perceived in the Yle task while the largest difference was with the Toxicity challenge data, FinSBERT leading the scoreboard by 0.09 points difference

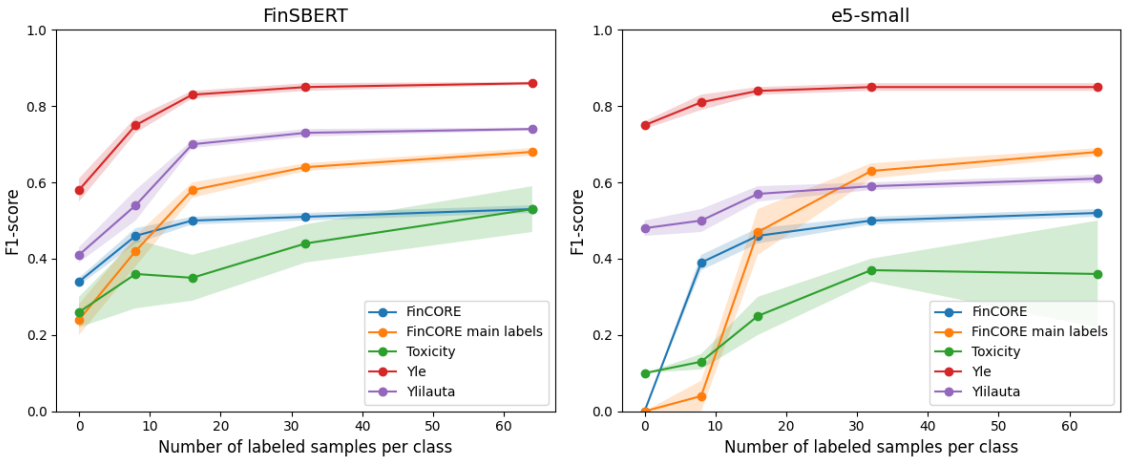

Figure 5.1: Comparison of monolingual FinSBERT and multilingual e5-small performances when fine-tuned with SetFit, highlighted area represents standard deviation

to the second-best model. From the multilingual models, e5-small seems promising, but fails in the Toxicity challenge task. Despite this, it performed well and on-par with FinSBERT in the complex multilabel FinCORE task. In Figure 5.1, you can find a comparison of these two models' performance across the different tasks. FinSBERT clearly fares better, especially in tasks that contain informal language, such as the Toxicity challenge dataset or the Ylilauta corpus.

Another point of interest was whether there would be a difference in the SetFit performance with multilabel and multiclass classification tasks, the hypothesis being that multiclass tasks would perform better than multilabel tasks. Using this method of fine-tuning, multilabel tasks such as FinCORE and the Toxicity challenge clearly receive worse results than multiclass tasks such as Yle and Ylilauta corpora. However, there was some variation, and it would seem that the quality or format of the language also has an effect. The more neutral FinCORE and Yle tasks received overall better scores than the more informal Toxicity and Ylilauta tasks, regardless of task composition. Even though SetFit's performance in the multilabel tasks was somewhat lacking, it could be argued that it might work as an alternative solution in multiclass labelling, even if state-of-the-art results are not reached.

This being a thesis about few-shot classification, how much data is needed then? Even though SetFit is aimed for few-shot classification, according to the results found in this research, in general, more data leads to better performance. Previous research has seen success with larger datasets [7], [14], [16], [17], [60], and Tunstall et al. recommend increasing the amount of data rather than training time for performance improvements [13]. However, while looking at Figure 5.1, an elbow in the learning curve can be observed when increasing the amount of data, and the performance increase decreases after 32 or even 16 examples. As shown in Figure 5.1, certain tasks seem more data-hungry than others. Yle performance is quite stable already with 16 examples per label, whereas the Toxicity challenge task learning curves with the two models presented are contradictory: looks like FinSBERT might have still learnt something with a larger dataset, but e5-small performance has already dropped after increasing the number of examples per label to 64 (please do note the large standard deviation). The sales pitch of "only 8 examples per label" [13] doesn't seem to hold at least with these example tasks — although if 16 or 32 examples per label is what it takes to achieve decent results, it isn't that far either. This number is considerably lower than the sheer amount of data used in fine-tuning the benchmark models.

In addition to fewer data requirements, the other benefit of SetFit is the lighter computational resource needs compared to many other fine-tuning methods. For example, fine-tuning with the seven-label Toxicity challenge dataset with 16 samples per label, the training time averages in 57.4 seconds per instance with MiniLM and 1.94 minutes with the larger XLM-R model. The largest compute time turned out to be in the e5-small model, but at least with the training set sizes smaller than 32 examples per label, the difference to the other models isn't too large. MiniLM is the fastest to train overall and might be worth a consideration in situations where the available computational resources are severely limited.

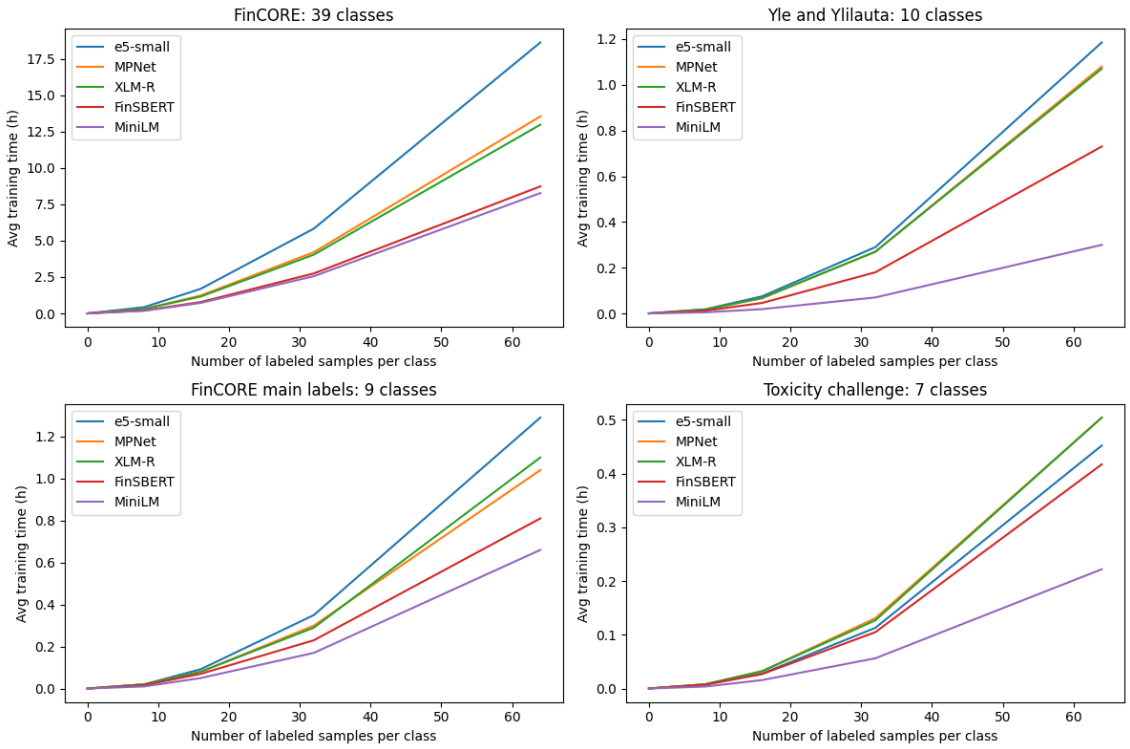

Figure 5.2: SetFit average training times with different numbers of classes and samples per label

Due to SetFit's data augmentation feature, training time grows relatively quickly in relation to the number of labels and the number of samples per label. Although training with the smallest dataset size was very fast with all the tasks, the time increased quite a bit when growing the amount of data, as evidenced in Figure 5.2. This could probably be countered by optimising the number of learning steps, which could also be beneficial to reduce overfitting. In this research, I did not utilise early stopping callback to return the best model but trained each instance for one full epoch like Tunstall et al. [13]. Another options to reduce training time could be downsampling the number of examples by for example manually curating the training sets, such as in [8], or setting SetFit to generate a smaller number of sentence pairs for training.

There doesn't seem to be a trend towards favouring larger models, rather the

small, 125M parameter monolingual FinSBERT model fares the best in all of the tasks, and the second-place e5-small is even smaller than that with only 118M parameters. The MPNet and XLM-R models, despite being over double the size, fall short in performance. The size difference between the base model FinBERT and the Sentence Transformer FinSBERT isn't very large, so in order to reach best performance with the datasets used in this thesis, fine-tuning FinBERT instead of FinSBERT is advisable. As I did not test fine-tuning FinBERT in a few-shot setting, the two method's performance cannot be compared in a straightforward manner. In the case where the data is sparse, it might be worth it to test fine-tuning a Sentence Transformer model with SetFit.

All models were trained for one full epoch, but the authors of [13] suggest that shorter training might yield better results and reduce overfitting. This is promising for lower-end devices and would reduce the computational resources needed for training even further. However, testing this hypothesis is beyond the scope of this thesis and would make interesting research in the future.

In this research, in most of the tasks, the difference to the state-of-the-art results remained quite large. All the Sentence Transformer models were fine-tuned for one full epoch with a stable learning rate. The original authors of [13] used a different learning rate of 1e-3, but I opted for the library default of 2e-5. At the time of doing my own experiments, some of the features in the experimental setup in [13] were already deprecated from the SetFit library. Such is the case of the sampling strategy, which I chose to optimise based on the FinCORE task performance and train time with FinSBERT. This differs from the original paper since the library itself had already deprecated the original sampling method. This might mean that the results are not necessarily comparable to the ones in the original paper and that with further parameter optimisation, better results might have been obtained. It could be interesting to, for example, choose a smaller sample size such as 16 samples

per label and optimise learning rate and training steps based on that.

Choosing the training sets manually to be representative in the manner of [8] could also lead to better results. In this experiment, the training sets were randomised and might have been inconsistent in quality. However, there is no indication of this when comparing the performance across different models: there are no particular training sets that consistently yield better or worse results than the others. In general, the results were surprisingly stable. Of course, in a data-sparse scenario, it might make sense to ensure the quality of each labelled example. In the case of FinCORE, I chose to include hybrid examples with several labels in the dataset. This could have been avoided, choosing only examples that represent a single main register and a possible sub-register. The same can be said for the Toxicity challenge dataset, where it would have been possible to choose only the "purest" representations of each class, with a minimum number of coinciding labels. However, by doing the experiments in the current way, it might have assisted the Sentence Transformer fine-tuning by bringing often co-occurring categories closer together in the vector space. This could also be one reason for the strange preference of the Insult+Obscene+Toxicity combination in the Toxicity challenge task.

6 Conclusion

In this thesis, I set out to find if fine-tuning Sentence Transformer models with SetFit could achieve few-shot text classification performance on par with the state-of-the-art results in Finnish and also understand the capabilities of the method by comparing monolingual and multilingual models and a range of text classification tasks, including multilabel and multiclass classification.

My first research question was: "How well do the fine-tuned Sentence Transformer models perform with Finnish tasks compared to the reported benchmarks? Can the SetFit method reach results comparable to the state-of-the-art?" and my initial hypothesis was that following the results by Tunstall et al. [13], state-of-the-art results would not be achieved. This hypothesis was proven correct: I was unable to reach previous state-of-the-art results with SetFit. However, the results are promising for data-sparse classification tasks and lower-end computing: one can achieve decent classification results with a small amount of data and few computational resources.

The second research question was: "Is there a difference in performance when fine-tuning Finnish Sentence Transformer models versus multilingual Sentence Transformer models with Finnish data?" and I found that the SetFit performance was linked to the model selection. The monolingual 125M parameter FinSBERT was generally the best in all tasks, followed by the multilingual e5-small. This indicates that the monolingual Sentence Transformer has an advantage in this kind of few-

shot learning, and that bigger models do not necessarily yield better results with this method, some of the models being double the size of the best performing ones.

The third and final research question was: "Is there a difference in performance when fine-tuning Sentence Transformers with multilabel classification versus multiclass classification in Finnish?" and my hypothesis was that SetFit would succeed better in multiclass tasks than in multilabel tasks. The hypothesis was proven true, as the SetFit method showed promise with multiclass tasks, nearing state-of-the-art results with considerably smaller datasets than in the original benchmarks of the Yle and Ylilauta tasks in Virtanen et al. [2]. With SetFit, the Yle task gained a maximum F1-score of 0.86 (compared to the originally reported benchmark of 0.92 accuracy), and with the Ylilauta dataset, a maximum F1-score of 0.74 was reached (compared to the original benchmark of 0.82 accuracy). However, with the multilabel FinCORE and Toxicity challenge tasks, the results left much to be desired. This is especially the case with the Toxicity challenge dataset, where each model appeared to overfit to favour certain label combinations.

Even though the results of this research show promise, any generalisations must be taken with a grain of salt. For now, there is no comparison with fine-tuning the state-of-the-art base models with the corresponding size training data sets, meaning that any straight comparison with them and SetFit cannot be made. The results from this research also could be improved: for comparison's sake, the training arguments were not optimised, and the models might have achieved better results with further optimisation. I also did not compare SetFit to other few-shot methods, and there might be other methods more suitable for situations when the data is limited but the computational resources are not. As state-of-the-art results were not reached in any of the tasks tested, it might also make sense just to fine-tune e.g. a FinBERT Transformer model when there is enough data available.

This thesis laid some basic groundwork for validating few-shot methods in Finnish

language text classification. In the future, there are many paths to explore in few-shot learning research. For example, it could be very interesting to see SetFit used to its full capabilities with proper parameter optimisation by, for example, restricting the training data set size and optimising other parameters, such as learning rate and the number of training steps. Another thing to consider would be the use of more curated datasets. In this thesis, the data here was randomly sampled, and might not have been the best representation. According to [8], samples chosen by a human expert yielded good results. For such small datasets, this wouldn't be an impossible task. A third possible option for future research would be to compare SetFit with other few-shot methods and possibly test out the Finnish benchmarking dataset introduced in [18] to gain understanding of SetFit's performance compared to other existing methods.

In the age of GenAI, one can easily generate any number of examples with just a prompt. While this kind of synthetically generated data is certainly a possibility, there still remain fields of research a generative model might not have been exposed to in greater lengths. Such is the case with many high security domains, as well as highly specific research areas. In addition to this, one must take into account the possible bias, possible inaccuracies and other limitations when using machine-generated data. With the constant advances in the field of NLP, the situation might, however, change quite rapidly. At the moment of writing this thesis, SetFit can be considered as a cost-efficient solution for Finnish text classification especially when the data is scarce and on-premise processing is needed.

References

[1] S. Subramanian, V. Elango, and M. Gungor, *Small language models (slms) can still pack a punch: A survey*, 2025. arXiv: 2501.05465 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2501.05465.

[2] A. Virtanen, J. Kanerva, R. Ilo, *et al.*, "Multilingual is not enough: BERT for Finnish", *CoRR*, vol. abs/1912.07076, 2019. arXiv: 1912.07076. [Online]. Available: http://arxiv.org/abs/1912.07076.

[3] V. Kortesalmi, "Sentiment analysis with language models on finnish workplace well-being surveys", M.S. thesis, University of Helsinki, 2024. [Online]. Available: http://hdl.handle.net/10138/577817.

[4] S. Samsi, D. Zhao, J. McDonald, *et al.*, *From words to watts: Benchmarking the energy costs of large language model inference*, 2023. arXiv: 2310.03003 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2310.03003.

[5] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP", in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3645–3650. DOI: 10.18653/v1/P19-1355. [Online]. Available: https://aclanthology.org/P19-1355/.

[6] A. Boulemtafes, A. Derhab, and Y. Challal, "A review of privacy-preserving techniques for deep learning", *Neurocomputing*, vol. 384, pp. 21–45, 2020, ISSN:

0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.11.041. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219316431.

[7] M. Loerakker, L. Müter, and M. Schraagen, "Fine-tuning language models on Dutch protest event tweets", in *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024)*, A. Hürriyetoğlu, H. Tanev, S. Thapa, and G. Uludoğan, Eds., St. Julians, Malta: Association for Computational Linguistics, 2024, pp. 6–23. [Online]. Available: https://aclanthology.org/2024.case-1.2/.

[8] L. Loukas, I. Stogiannidis, P. Malakasiotis, and S. Vassos, "Breaking the bank with ChatGPT: Few-shot text classification for finance", in *Proceedings of the Fifth Workshop on Financial Technology and Natural Language Processing and the Second Multimodal AI For Financial Forecasting*, C.-C. Chen, H. Takamura, P. Mathur, R. Sawhney, H.-H. Huang, and H.-H. Chen, Eds., 2023, pp. 74–80. [Online]. Available: https://aclanthology.org/2023.finnlp-1.7/.

[9] V. Skantsi and V. Laippala, "Analyzing the unrestricted web: The finnish corpus of online registers", *Nordic Journal of Linguistics*, vol. 1, no. 1, 2023, ISSN: 15024717. DOI: 10.1017/S0332586523000021.

[10] V. Laippala, J. Egbert, D. Biber, and A.-J. Kyröläinen, "Exploring the role of lexis and grammar for the stable identification of register in an unrestricted corpus of web documents", *Language Resources and Evaluation*, vol. 55, no. 3, pp. 757–788, Sep. 2021, ISSN: 1574-020X. DOI: 10.1007/s10579-020-09519-z. [Online]. Available: https://doi.org/10.1007/s10579-020-09519-z.

[11] B. Kilic, F. Bex, and A. Gatt, "Contrast is all you need", in *ASAIL 2023 - Automated Semantic Analysis of Information in Legal Text*, F. Lagioia, J.

Mumford, D. Odekerken, and H. Westermann, Eds., ser. CEUR Workshop Proceedings, CEUR, 2023, pp. 72–82.

[12] A. Eskelinen, L. Silvala, F. Ginter, S. Pyysalo, and V. Laippala, "Toxicity detection in Finnish using machine translation", in *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, T. Alumäe and M. Fishel, Eds., Tórshavn, Faroe Islands: University of Tartu Library, May 2023, pp. 685–697. [Online]. Available: https://aclanthology.org/2023.nodalida-1.68.

[13] L. Tunstall, N. Reimers, U. E. S. Jo, *et al.*, "Efficient Few-Shot Learning Without Prompts", 2022. arXiv: 2209.11055. [Online]. Available: http://arxiv.org/abs/2209.11055.

[14] A. S. Kwak, C. Jeong, J. W. Lim, and B. Min, *A korean legal judgment prediction dataset for insurance disputes*, 2024. arXiv: 2401.14654 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2401.14654.

[15] T. Hadeliya and D. Kajtoch, *Evaluation of few-shot learning for classification tasks in the polish language*, 2024. arXiv: 2404.17832 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2404.17832.

[16] K. Pannerselvam, S. Rajiakodi, S. Thavareesan, S. Thangasamy, and K. Ponnusamy, "SetFit: A robust approach for offensive content detection in Tamil-English code-mixed conversations using sentence transfer fine-tuning", in *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, B. R. Chakravarthi, R. Priyadharshini, A. K. Madasamy, *et al.*, Eds., St. Julian's, Malta: Association for Computational Linguistics, 2024, pp. 35–42. [Online]. Available: https://aclanthology.org/2024.dravidianlangtech-1.6/.

[17] V. Beliveau, H. Kaas, M. Prener, *et al.*, *Classification of radiological text in small and imbalanced datasets in a non-english language*, 2024. arXiv: 2409.20147 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2409.20147.

[18] R. Luukkonen, V. Komulainen, J. Luoma, *et al.*, "FinGPT: Large generative models for a small language", in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 2710–2726. DOI: 10.18653/v1/2023.emnlp-main.164. [Online]. Available: https://aclanthology.org/2023.emnlp-main.164/.

[19] S. Rönnqvist, V. Skantsi, M. Oinonen, and V. Laippala, "Multilingual and Zero-Shot is Closing in on Monolingual Web Register Classification", *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pp. 157–165, 2021. [Online]. Available: https://aclanthology.org/2021.nodalida-main.16.

[20] W. Mcculloch and W. Pitts, "A logical calculus of ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, vol. 5, pp. 127–147, 1943.

[21] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain", eng, *Psychological review*, vol. 65, no. 6, pp. 386–408, 1958, ISSN: 0033-295X. DOI: 10.1037/h0042519.

[22] D. Jurafsky and J. H. Martin, *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, N.J.: Pearson Prentice Hall, 2009.

[23] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks", *CoRR*, vol. abs/1908.10084, 2019. arXiv: 1908.10084. [Online]. Available: http://arxiv.org/abs/1908.10084.

[24] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network", *CoRR*, vol. abs/1808.03314, 2018. arXiv: 1808.03314. [Online]. Available: http://arxiv.org/abs/1808.03314.

[25] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention Is All You Need", *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. Nips, pp. 5999–6009, Jul. 2017, ISSN: 10495258. arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762.

[26] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language models are few-shot learners*, 2020. arXiv: 2005.14165 [cs.CL].

[27] G. Yenduri, R. M, C. S. G, *et al.*, *Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions*, 2023. arXiv: 2305.10435 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2305.10435.

[28] OpenAI, J. Achiam, S. Adler, *et al.*, *Gpt-4 technical report*, 2024. arXiv: 2303.08774 [cs.CL].

[29] B. Workshop, : T. L. Scao, *et al.*, *Bloom: A 176b-parameter open-access multilingual language model*, 2023. arXiv: 2211.05100 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2211.05100.

[30] H. Touvron, T. Lavril, G. Izacard, *et al.*, *Llama: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2302.13971.

[31] J. Kanerva, F. Ginter, L. H. Chang, *et al.*, "Towards diverse and contextually anchored paraphrase modeling: A dataset and baselines for Finnish", *Natural Language Engineering*, vol. 34, no. 6, pp. 1–35, 2023, ISSN: 14698110. DOI: 10.1017/S1351324923000086.

[32] A. Conneau, K. Khandelwal, N. Goyal, *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale", *CoRR*, pp. 31–38, Nov. 2019. DOI: 10.18653/v1/p19-4007. arXiv: 1911.02116. [Online]. Available: http://arxiv.org/abs/1911.02116.

[33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, no. Mlm, pp. 4171–4186, Oct. 2018. arXiv: 1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781.

[35] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation", in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162.

[36] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information", *Transactions of the Association for Computational Linguistics*, vol. 5, L. Lee, M. Johnson, and K. Toutanova, Eds., pp. 135–146, 2017. DOI: 10.1162/tacl_a_00051. [Online]. Available: https://aclanthology.org/Q17-1010/.

[37] R. Kiros, Y. Zhu, R. Salakhutdinov, *et al.*, *Skip-thought vectors*, 2015. arXiv: 1506.06726 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1506.06726.

[38] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data", in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa, and S. Riedel, Eds., Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 670–680. DOI: 10.18653/v1/D17-1070. [Online]. Available: https://aclanthology.org/D17-1070/.

[39] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference", in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. Màrquez, C. Callison-Burch, and J. Su, Eds., Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 632–642. DOI: 10.18653/v1/D15-1075. [Online]. Available: https://aclanthology.org/D15-1075/.

[40] D. Cer, Y. Yang, S. Kong, *et al.*, "Universal Sentence Encoder", *CoRR*, vol. abs/1803.11175 2018. arXiv: 1803.11175. [Online]. Available: http://arxiv.org/abs/1803.11175.

[41] Y. Liu, M. Ott, N. Goyal, *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach", no. 1, 2019, ISSN: 2331-8422. arXiv: 1907.11692. [Online]. Available: http://arxiv.org/abs/1907.11692.

[42] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference", in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, M. Walker, H. Ji, and A. Stent, Eds., New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 1112–1122. DOI: 10.18653/v1/N18-1101. [Online]. Available: https://aclanthology.org/N18-1101/.

[43] A. Edwards and J. Camacho-Collados, "Language models for text classification: Is in-context learning enough?", in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, Eds., Torino, Italia: ELRA and ICCL, May 2024, pp. 10 058–10 072. [Online]. Available: https://aclanthology.org/2024.lrec-main.879/.

[44] V. Laippala, R. Kyllönen, J. Egbert, D. Biber, and S. Pyysalo, "Toward Multilingual Identification of Online Registers", in *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, Turku, Finland: Linköping University Electronic Press, 2019, pp. 292–297. [Online]. Available: https://www.aclweb.org/anthology/W19-6130.

[45] M. Tolba, S. Ouadfel, and S. Meshoul, "Hybrid ensemble approaches to online harassment detection in highly imbalanced data", *Expert Systems with Applications*, vol. 175, p. 114 751, 2021, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.114751. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421001925.

[46] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, *Dice loss for data-imbalanced nlp tasks*, 2020. arXiv: 1911.02855 [cs.CL].

[47] S. Shaikh, S. M. Daudpota, A. S. Imran, and Z. Kastrati, "Towards improved classification accuracy on highly imbalanced text dataset using deep neural language models", *Applied Sciences*, vol. 11, no. 2, 2021, ISSN: 2076-3417. DOI: 10.3390/app11020869. [Online]. Available: https://www.mdpi.com/2076-3417/11/2/869.

[48] S. Wang, H. Fang, M. Khabsa, H. Mao, and H. Ma, *Entailment as few-shot learner*, 2021. arXiv: 2104.14690 [cs.CL].

[49] M. Lewis, Y. Liu, N. Goyal, *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. [Online]. Available: https://aclanthology.org/2020.acl-main.703/.

[50] T. Schick and H. Schütze, *Exploiting cloze questions for few shot text classification and natural language inference*, 2021. arXiv: 2001.07676 [cs.CL].

[51] J. Luoma, M. Oinonen, M. Pyykönen, V. Laippala, and S. Pyysalo, "A broad-coverage corpus for Finnish named entity recognition", *Proceedings of the 12th Language Resources and Evaluation Conference*, no. 5, pp. 4615–4624, 2020. [Online]. Available: https://aclanthology.org/2020.lrec-1.567.

[52] J. Luotolahti, J. Kanerva, V. Laippala, S. Pyysalo, and F. Ginter, "Towards universal web parsebanks", in *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, J. Nivre and E. Hajičová, Eds., Uppsala, Sweden: Uppsala University, Uppsala, Sweden, Aug. 2015, pp. 211–220. [Online]. Available: https://aclanthology.org/W15-2124/.

[53] J. Kanerva, F. Ginter, N. Miekka, A. Leino, and T. Salakoski, "Turku Neural Parser Pipeline: An End-to-End System for the CoNLL 2018 Shared Task", in *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 133–142, ISBN: 9781948087827. DOI: 10.18653/v1/K18-2013. [Online]. Available: http://aclweb.org/anthology/K18-2013.

[54] C. S. Perone, R. P. Silveira, and T. S. Paula, "Evaluation of sentence embeddings in downstream and linguistic probing tasks", *CoRR*, vol. abs/1806.06259, 2018. arXiv: 1806.06259. [Online]. Available: http://arxiv.org/abs/1806.06259.

[55] G. Piao, "Scholarly text classification with sentence bert and entity embeddings", in *Trends and Applications in Knowledge Discovery and Data Mining*, M. Gupta and G. Ramakrishnan, Eds., Cham: Springer International Publishing, 2021, pp. 79–87, ISBN: 978-3-030-75015-2.

[56] D. Tam, R. R. Menon, M. Bansal, S. Srivastava, and C. Raffel, "Improving and simplifying pattern exploiting training", in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4980–4991. DOI: 10.18653/v1/2021.emnlp-main.407. [Online]. Available: https://aclanthology.org/2021.emnlp-main.407/.

[57] R. Karimi Mahabadi, L. Zettlemoyer, J. Henderson, *et al.*, "Prompt-free and efficient few-shot learning with language models", in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3638–3652. DOI: 10.18653/v1/2022.acl-long.254. [Online]. Available: https://aclanthology.org/2022.acl-long.254/.

[58] H. Liu, D. Tam, M. Muqeeth, *et al.*, *Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning*, 2022. arXiv: 2205.05638 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2205.05638.

[59] P. Keung, Y. Lu, G. Szarvas, and N. A. Smith, "The multilingual Amazon reviews corpus", in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 4563–4568. DOI: 10.18653/v1/2020.emnlp-main.369. [Online]. Available: https://aclanthology.org/2020.emnlp-main.369/.

[60] E. Fsih, S. Kchaou, R. Boujelbane, and L. Hadrich-Belguith, "Benchmarking transfer learning approaches for sentiment analysis of Arabic dialect", in *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, H. Bouamor, H. Al-Khalifa, K. Darwish, *et al.*, Eds., Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, 2022, pp. 431–435. DOI: 10.18653/v1/2022.wanlp-1.44. [Online]. Available: https://aclanthology.org/2022.wanlp-1.44/.

[61] N. Reimers and I. Gurevych, *Making monolingual sentence embeddings multilingual using knowledge distillation*, 2020. arXiv: 2004.09813 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2004.09813.

[62] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. A. Smith, "Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping", *CoRR*, vol. abs/2002.06305, 2020. arXiv: 2002.06305. [Online]. Available: https://arxiv.org/abs/2002.06305.

[63] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, and Y. Artzi, "Revisiting few-sample BERT fine-tuning", *CoRR*, vol. abs/2006.05987, 2020. arXiv: 2006.05987. [Online]. Available: https://arxiv.org/abs/2006.05987.

[64] Ylilauta, *The Downloadable Version of the Ylilauta Corpus*, data set, 2016. [Online]. Available: http://urn.fi/urn:nbn:fi:lb-2016101210.

[65] Yleisradio, *Yle Finnish News Archive 2011-2018, source*, data set. [Online]. Available: http://urn.fi/urn:nbn:fi:lb-2017070501.

[66] L. Repo, V. Skantsi, S. Rönnqvist, *et al.*, "Beyond the English web: Zero-shot cross-lingual and lightweight monolingual classification of registers", in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, I.-T. Sorodoc, M. Sushil, E. Takmaz, and E. Agirre, Eds., Online: Association for Computational Linguistics, Apr. 2021, pp. 183–191. DOI: 10.18653/v1/2021.eacl-srw.24. [Online]. Available: https://aclanthology.org/2021.eacl-srw.24.

Appendix A FinCORE Register Distribution

Register name	Abbreviation	N
Narrative (main)	**NA**	**3 956**
News reports / news blogs	NE	1 359
Personal blog	PB	1 160
Community blog	CB	374
Sports reports	SR	357
Magazine / Online article	OA	342
Story	FC	121
Travel blog	TB	82
Historical article	HA	79
Informational description (main)	**IN**	**1 719**
Description of a thing	DT	550
Encyclopedia articles	EN	238
Description of a person	DP	142
Information blog	IB	125
Report	RP	121
Legal terms / conditions	LT	114
Research article	RA	78
Course material	CM	61

Job description	JD	47
FAQs	FA	23
Opinion (main)	**OP**	**1 399**
Reviews	RV	554
Religious text/sermon	RS	405
Opinion blog	OB	363
Advice	AV	31
Machine translated / generated texts (main)	**MT**	**1 388**
Informational persuasion (main)	**IP**	**1 334**
Description with intent to sell	DS	1 145
News-opinion blog / editorial	EB	97
Interactive discussion (main)	**ID**	**1 081**
Discussion forums	DF	749
Question-answer forum	QA	91
How-to / instructions (main)	**HI**	**549**
Recipe	RE	45
Spoken (main)	**SP**	**75**
Interview	IT	50
Formal speech	FS	25
Lyrical (main)	**LY**	**25**
Poem	PO	25