DOCTORAL THESIS

Improving Multilingual Automatic Cyberbullying Detection With Feature Density And Cross-lingual Zero-shot Transfer

素性密度及びクロスリンガルゼロショット転移学習による多言語のネットいじめ自動検出の改良に関する研究

by

Juuso Kalevi Kristian Eronen

Advised by:
Michal Ptaszynski
Fumito Masui
Takashi Okumura

KITAMI INSTITUTE OF TECHNOLOGY
GRADUATE SCHOOL OF ENGINEERING

September 2022

Acknowledgements

This work would not have been possible without all the support that I received. I would like to reserve this page to thank them.

First and foremost, I want to thank my supervisors, Professors Michal Ptaszynski and Fumito Masui, for their invaluable assistance and insights, and for providing support and feedback which helped me a lot in all aspects during my course. I am deeply indebted for their extensive support and mentorship in my academic development and served as a constant source of expertise and encouragement throughout my studies. Furthermore, I would like to thank them for introducing me to the field of Natural Language processing when we first met, which gave me the possibility of choosing this career path.

I also thank the members of my dissertation committee, Professor Takashi Okumura, Professor Yasunari Maeda and Professor Yoshio Abe for offering their efforts in giving valuable feedback and comments to improve this thesis.

Lastly, I would like to say a heartfelt thank you to my family and friends, who endured this long process with me, always offering help and support. I would like to give special thanks to my girlfriend, for her encouragement and understanding.

Juuso Eronen
September 2022

ABSTRACT

In this thesis, I study two different methods for improving multilingual automatic cyberbullying detection. First, I study the effectiveness of Feature Density (FD) using different linguistically-backed feature preprocessing methods in order to estimate dataset complexity, which in turn is used to comparatively estimate the potential performance of machine learning (ML) classifiers prior to any training. I hypothesize that estimating dataset complexity allows for the reduction of the number of required experiments iterations, making it possible to optimize the resource-intensive training of ML models which is becoming a serious issue due to the increases in available dataset sizes and the ever rising popularity of models based on Deep Neural Networks (DNN). The problem of constantly increasing needs for more powerful computational resources is also affecting the environment due to alarmingly-growing amount of CO2 emissions caused by training of large-scale ML models. I use cyberbullying datasets collected for multiple languages, namely English, Japanese and Polish. The difference in linguistic complexity of datasets allows me to additionally discuss the efficacy of linguistically-backed word preprocessing.

Second, I study the selection of transfer languages for automatic abusive language detection. I demonstrate the effectiveness of cross-lingual transfer learning for zero-shot abusive language detection. This way it is possible to use existing data from higher-resource languages to build better detection systems for languages lacking data. The datasets are from eight different languages from three language families. I measure the distance between the languages using several language similarity measures, especially by quantifying the World Atlas of Language Structures. I show that there is a correlation between linguistic similarity and classifier performance, making it possible to choose an optimal transfer language for zero shot abusive language detection.

Next, I demonstrate that this method is also generally applicable to multiple Natural Language Processing tasks, specifically sentiment analysis, named entity recognition and dependency parsing. I show that there is also a correlation between linguistic similarity and zero-shot cross-lingual transfer performance for these tasks, allowing me to select an ideal transfer language in order to aid with the problem of dealing with languages that do not currently have a sufficient amount of data. Lastly, I show that the World Atlas of Language Structures can be quantified into an effective linguistic similarity method.

ABSTRACT IN JAPANESE (論文内容の要旨)

本論文では，多言語におけるネットいじめの自動検出の性能を向上させるため，2つの方法について検討を行う．まず，データセットの複雑性を推定するために，特徴量となる言語 学的素性(そせい)を用いた特徴密度(Feature Density, 以下:FD,素性密度とも)の有効性について研究する．FDは，学習前の機械学習（ML）分類器の潜在性能を対照的に想定するために用いられる．近年，利用可能な公開データセットの数及びサイズが増加じ，さらにディープニューラルネットワーク（DNN）に基づくモデルの人気の上昇により，最適なモデルを選抜するのに多大な実験の反復回数を繰り返さなければならないことが深刻な問題となっている．データセットの複雑性を推定することで，必要な実験の反復回数を減らすことができ，リソース集約的なMLモデルの学習を最適化できることを仮設する．さらに，大規模なMLモデルの学習によって発生するCO2排出量が増加し，環境にも影響を与えていることが指摘されている．実験では，英語，日本語，ポーランド語という多言語で収集されたネットいじめのデータセットを使用し，データセットの言語的複雑さの違いに伴い，単語の言語学的前処理方法の有効性について考察を行う．

次に，ネットいじめの自動検出のための移転言語の選択について調査を行う．具体的には，ゼロショット多言語間ネットいじめの自動検出における言語間移転学習の有効性の実証を行う．ゼロショット多言語間移転学習では，多資源の言語において収集された既存のデータを用いて，データが不足している少資源の言語向けのより良い検出システムを開発することが可能である．データセットは3つの言語族からなる8つの言語からなるものである．いくつかの言語類似性指標を用い，特にWorld Atlas of Language Structuresを定量化することで言語間の距離を測定する．言語類似度と分類器の性能の間に相関があることを示し，ゼロショットでのネットいじめ検出に最適な移転言語を選択することが可能であることを示す．

そして，移転学習用の最適な言語選抜手法を複数の自然言語処理の課題，特に感情極性解析，固有表現抽出，構文解析に一般的に適用できることを示す．また，これらの課題において，言語類似性指標とゼロショット多言語間移転学習の性能の間に相関があることを示し，十分なデータ量がない言語を扱う問題を支援するために，最適な移転言語を選択することができることを示す．最後に，World Atlas of Language Structuresが効果的な言語類似性計算手法に定量化できることを示す．

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AI	artificial intelligence
BERT	Bidirectional Encoder Representations from Transformers
BoW	bag of words
CB	cyberbullying
CHNK	chunking
CNN	convolutional neural network
DEP	dependency parsing
DNN	deep neural network
FD	feature density
IP	internet patrol
kNN	k-nearest neighbor
LAS	label attachment score
LD	lexical density
ML	machine learning
NB	naive bayes
NER	named entity recognition
NLP	natural language processing
POS	parts-of-speech
SNS	social networking service
SO-PMI-IR	Semantic Orientation from Pointwise Mutual Information and Information Retrieval
SVM	support vector machine
TF-IDF	term frequency-inverse document frequency
WALS	World Atlas of Language Structures
XLM	Cross-lingual Language Model

Chapter 1

Introduction

The rise of communication between people through Internet during the last decades has brought massive amounts of information to the reach of everyone and forever changed the ways we communicate through instant messaging and social media. Unfortunately, this has not come without problems brought by the anonymity and openness.

The negative and abusive behaviour encountered online, known as cyberbullying (CB), is defined as the exploitation of open online means of communication, such as Internet forum boards, or social network services (SNS) to convey harmful and disturbing information about private individuals, often children and students [1]. Users' realization of the anonymity of online communications is one of the factors that make this activity attractive for bullies since they rarely face consequences of their improper behavior. The problem was further exacerbated by the popularization of smartphones and tablet computers that enable almost continuous usage of SNS anywhere, at home, work/school or in motion [2].

Messages that can be identified as cyberbullying usually ridicule someone's personality, body type or appearance, or include slandering or spreading rumors about the individual. This may drive its victims to even as far as self-mutilation or suicide, or, on the contrary, to a retaliation assault on their perpetrators [3]. Global spike in cyberbullying cases[1] opened a world wide discussion about whether such messages should be identified early to deter harm, and on freedom of speech on the Internet.

[1] `https://cyberbullying.org/summary-of-our-cyberbullying-research`

Harmful language in online communication can cause serious consequences to its victims. In the worst cases, it can lead to self-mutilation or suicide, or, on the contrary, to a retaliation assault on their perpetrators [3]. There have been multiple attempts to automate the detection of offensive content online [4, 5, 6] in order to reduce the human effort needed in prevention of the uncontrolled spread of harmful content on social media. Even though there are thousands of languages used in different social media platforms, the research on the detection of harmful content has only been done with a handful of them, mostly in English [7, 8], Japanese [9, 10], Polish [11], Arabic [12] and Hindi [13].

In certain countries, such as in Japan, the issue has been severe enough to be seen at ministerial level [14]. As one of the ways to solve the issue, Internet Patrol (IP) consisting of school workers has begun to track online forum pages and SNS featuring cyberbullying content. Unfortunately, as IP is carried out manually, reading through vast numbers of websites and SNS content makes it an uphill battle. To aid in this struggle, some research have started to develop methods for automatic detection of CB [9, 10, 8, 15].

Even with numerous improvements, the findings have unfortunately remained only slightly satisfactory. This is due to the plethora of language and vocabulary ambiguities and styles used in CB. Solving the problem of CB has become even more crucial after introducing global policies such as General Data Protection Regulation (GDPR) in EU[2], which put the weight of spotting and weeding out online harassment on the platforms themselves. Therefore the process of efficient implementing of automatic cyberbullying detection for different languages and social networking sites is one of the current most burning problems, which could greatly benefit from a method allowing to loosely approximate which classifier configurations can be rejected without the experimental process.

In order to contribute more to solving this problem, I am performing an in-depth study of the efficacy of the notion of Feature Density (FD) previously proposed by Ptaszynski et al. [16] to comparatively estimate the efficiency of various ML classifiers prior to training. Additionally I evaluate the usefulness of numerous linguistically-backed feature preprocessing approaches, including lemmas,

[2] `https://edps.europa.eu/data-protection/data-protection/reference-library/anti-harassment-procedures_en`

Named Entity Recognition (NER) and dependency information-based features, in an application for automated cyberbullying detection.

To optimize the use of resources when dealing with NLP problems, it would be useful to be able to have information about the complexity of different datasets, meaning how difficult are the datasets for a classifier to learn and generalize upon. This problem has been recognized in other fields such as image recognition [17]. Even though natural language complexity has been studied through lexical [18, 19, 20, 21] and syntactic complexity [22, 23, 24, 25], mostly within the second language education field, there have been very few or no applications to use this kind of measures in estimating dataset complexity in ML tasks.

In addition, being able to detect harmful language like hate speech and cyberbullying (CB) also in languages lacking a the required training data would be a great aid, as social media is used in thousands of languages, of which only a small fraction have proper data for model development. It is also important to detect offensive content as urgently and effectively as possible because of its increasing prevalence and serious consequences [26]. Users' realization of the anonymity of online communications is one of the factors that make this activity attractive for harassers and bullies since they rarely face consequences of their improper behavior.

Recently, the research of automatic hate speech detection has expanded to dealing with low-resource languages. This has come with new challenges as these languages lack proper datasets to be used for training the detection models. To get around this problem, it has been shown that with cross-lingual transfer, the performance on languages lacking data can be improved by leveraging knowledge from other higher resource languages. This has also been demonstrated to be an effective technique in improving offensive content detection in low resource languages by using cross-lingual word embeddings and multilingual transformer models [27, 28, 29, 30].

However, choosing the optimal language for the transfer remains widely an understudied problem. Usually, it is up to the individual (researcher, or ML practitioner) to decide experimentally or by pure intuition which language might be suitable for the transfer, based on their field experience and accumulated theoretical knowledge. For example, one could select the transfer language by looking at languages belonging to the same language family as the target language [31]. But

this does not necessarily mean that the two languages would share the same linguistic features [32].

1.1 Organization

The reminder of this thesis is organized as follows. Chapter 2 describes the background of my research. First, I go through the previous research in the area of Abusive Language Identification with a focus on Cyberbullying Detection. Second, I go through research assessing some of the challenges the area currently faces, specifically, regarding the increasing computational requirements and model development for languages lacking data. In Chapter 3, I apply the concept of Feature Density, and explore its potential in comparatively estimating the efficiency of various ML classifiers prior to any training. Additionally I evaluate the usefulness of numerous linguistically-backed feature preprocessing approaches. In Chapter 4 I aim to answer the need of developing a method of selecting languages for cross-lingual transfer learning in order to aid the development of detection models for languages lacking proper training data. The approach to this problem is to study, whether different linguistic similarity metrics could be used for finding the optimal candidates for cross-lingual transfer. I also propose a novel multidomain linguistic similarity metric quantified from the World Atlas of Language Structures. In Chapter 5 I demonstrate that the method developed in the previous chapter can be generalized also to other NLP tasks. In Chapter 6 I summarize and review all of the principal findings of this research and discuss ideas for future work.

1.2 Contributions

One of the biggest challenges in the field of Cyberbullying detection is the urgency of the problem. The victimization rates of cyberbullying are constantly increasing due to the integration of Social Networking Services to daily life. This calls for a need to develop detection systems as fast and as efficiently as possible. However, this is being held back due to increases in computational costs in creating models and the cost of creating datasets. The goal of this research is trying to aid in tackling the issue of cyberbullying as fast and as efficiently as possible by finding

solutions to the ever increasing computational costs in creating detection tools and the lack of available datasets for most languages.

The research is conducted by studying the effectiveness of FD using different linguistically-backed feature preprocessing methods in order estimate dataset complexity, which in turn is used to comparatively estimate the potential performance of ML classifiers prior to any training. I hypothesise that FD will show correlations between various preprocessings and results, and by estimating dataset complexity, allows for the reduction of the number of required experiments iterations. This way it is possible to optimize the resource-intensive training of ML models which is becoming a serious issue due to the increases in available dataset sizes and the ever rising popularity of models based on Deep Neural Networks.

There are many classifiers and different ways to produce features that need to be considered when developing harmful and abusive behavior detection methods which is both time consuming and computationally intensive. Also, there are vast amounts of SNS platforms each of which are operating in one or multiple languages. It is practically impossible to develop a one-size-fits-all system for these platforms because of, for example, different user policies, like the definition of harmful content. It is difficult to deal with all languages at once due to the limitations of multilingual models still being widely unknown [33] and machine translation having its own issues with for example language specific semantics.

My approach concentrates on the classifiers and feature engineering. In practice, trying to estimate what kind of feature engineering methods would work best for different classifiers in different languages. This would make it possible to ignore feature engineering methods not viable for a particular classifier or language and only keep those that I believe could yield the highest performance without doing any actual training.

This would also answer the need of developing greener AI by ultimately reducing the CO_2 emissions caused by model training by proposing a method to estimate dataset complexity, and thus to comparatively estimate the potential performance of machine learning (ML) classifiers on a particular dataset prior to any training. The problem of constantly increasing needs for more powerful computational resources is affecting the environment due to alarmingly-growing amount of CO_2 emissions caused by training of large-scale ML models. The approach to this problem is to

find a way to train classifiers faster and more efficiently by decreasing the training load, which could be possible by looking at some general characteristics of a dataset, in this case, its complexity. This would allow for the reduction of the number of required experiments iterations.

Cyberbullying, while being a serious social problem, is also a very sophisticated problem from the point of view of linguistic representation. The difference in linguistic complexity of datasets makes it possible to additionally discuss the efficacy of novel linguistically-backed word embeddings. As the recent trends in NLP mostly focus on using words, like with BERT [34], there could be potential in preserving deeper relations between lexical items and structures, by for example including linguistic information like parts-of-speech or dependency information. In order to explore this potential, I propose to preserve the morphological, syntactic and other types of linguistic information by combining them with the pure tokens or lemmas

In addition, this research aims to deal with the problem of cyberbullying in a language that has no available training by utilizing cross-lingual transfer. However, the current methods for selecting languages for cross-lingual transfer learning are mainly based on the individual's own judgement based on their field experience and accumulated theoretical knowledge or simply choosing languages from the same language family [31]. The problem with the current selection methods are that they are completely unoptimized and prone to bias from the practitioner. In fact, one could argue that there is no systematic method that would give an actual score or ranking for the transfer language candidates.

The approach is to explore, whether different linguistic similarity metrics could be used for finding the optimal candidates for cross-lingual transfer. Supported by the findings of Gaikwad et al. [30], I hypothesize that linguistic similarity correlates with cross-lingual transfer efficacy, meaning that using more similar languages would yield a higher classification score. In practice, I fine tune cross-lingual pretrained language models, specifically mBERT and XLM-R, separately on each of the proposed languages (English, German, Danish, Polish, Russian, Japanese, Korean) and then perform zero-shot classification on the rest of the languages of the proposed set.

I propose to investigate the possibility that different linguistic similarity metrics

could be utilized when trying to find possible source language candidates for cross-lingual transfer also for other tasks than abusive language detection. I hypothesize that linguistic similarity correlates with cross-lingual transfer efficacy, meaning that by using more similar languages, a higher model performance would be achievable.

Also, in order to capture all aspects of a language, I propose a novel linguistic similarity metric quantified from the World Atlas of Language Structures (WALS). As WALS contains a variety of linguistic features from multiple domains such as phonological, grammatical and lexical, I hypothesize that it will perform better when selecting languages for cross-lingual transfer compared to existing similarity metrics.

Chapter 2

Background

2.1 Cyberbullying Detection

Even though the issue of CB has been researched in social sciences and child psychology for over ten years [1, 35], only a small number of significant attempts have been made so far to use information technology to help solve the problem. Here I introduce the most relevant studies up to the day.

For the first time, [9, 10] in 2010 performed affect analysis on a small CB dataset and discovered that the use of vulgar words were the distinctive features for CB. They trained an SVM classifier using a lexicon of such words and with multiple optimizations, they managed to detect CB with an F-score of 88.2%. However, as the amount of data increased, it caused a decrease in results, which caused the authors to abandon SVM as not ideal for language ambiguities typical for CB.

In other research, Sood et al. [36] focused on detecting personal insults and negative influence which could at most cause the Internet community to fall into recession, meaning if the harmful content would be left uncontrolled, people would start to leave the community. Their study used single words and bigrams as features, and weighted them using Boolean weighting (1/0), term frequency and TF-IDF. These were used to train an SVM classifier. Their dataset was a corpus collected from multiple online fora, totaling at six thousand entries. They used a crowd-sourcing approach (Mechanical Turk) with non-professional laypersons hired for the classification task to annotate the data.

Later, Dinakar et al. [8] introduced their method to detect and mitigate

cyberbullying. Their paper had a wider perspective, as they did not focus only on the detection of cyberbullying, but also included methods for mitigating the problem. This was an improvement compared to previous research. Their classifiers scored up to 58-77% of F-score in an English dataset. The results varied depending on the type of harassment they were attempting to classify. The best classifier they proposed was again SVM, which further confirms the effectiveness of SVMs for detecting cyberbullying, similarly to the research done in 2010 using a Japanese dataset [9].

An interesting work was done by Kontostathis et al. [7], who performed a thorough analysis of cyberbullying entries on Formspring.me. They identified usual cyberbullying patterns and used a machine learning method based on Essential Dimensions of Latent Semantic Indexing (EDLSI) to apply them in classification.

Cano et al. [37] introduced a Violence Detection Model (VDM), a weakly supervised Bayesian model. However, their focus was not strictly restricted on cyberbullying, but consisted of a more widened scope of generally understood “violence". This simplified the problem and made it more feasible for untrained annotators to work with. The datasets were extracted from violence-related topics on Twitter and DBPedia.

Nitta et al. [38] proposed a method extending Turney's SO-PMI-IR score [39] to automatically detect harmful entries. They used the score to calculate the relevance of a document with harmful contents. The seed words were grouped into three categories (abusive, violent, obscene) and the relevance of categories was maximized. The method was evaluated comparatively high as the best achieved Precision was around 91% (although with Recall less then 10%).

A re-evaluation of their method two years later unfortunately suggested that the method lost a major amount of its Precision (over 30 percentage-point drop) over the span of two years [15]. They hypothesized that this could be the cause of external factors like Web page re-ranking, or changes in SNS user policies, etc. The method was improved by acquiring and filtering new harmful seed words automatically with some success (P=76%), but they were unable to achieve results close to the original performance. Later an automatic method for the seed word acquisition [40] was developed with positive results. However, this method was deemed inefficient and impractical compared to a more direct machine learning

based approach with a properly annotated dataset.

Sarna et al. [4] based their method on a set of features like "bad words", positive/negative sentiment words, and other common features like pronouns, etc., to estimate user credibility. These features were applied to commonly used classifiers (Naive Bayes, kNN, Decision Trees, SVM). The obtained classification results were further used in User Behavior Analysis model (BAU), and User Credibility Analysis (CAU) model. Even though their approach included the use of phenomena such as irony, or rumors, in practice they unfortunately only focused on messages containing "bad words." Moreover, neither the words themselves, the dataset, nor its annotation schema were sufficiently described in the paper.

Ptaszynski et al. [41] suggested a pattern-based language modeling system. The patterns, identified as ordered combinations of sentence elements, were extracted with the use of a Brute-Force search-inspired algorithm and used for classification. They reported promising initial findings and further developed the system by adding several data pre-processing techniques [11].

Finally, Ptaszynski et al. [16] proposed a method of using Linguistically-backed preprocessing methods and increased Feature Density to find an optimal way to preprocess the data in order to achieve higher performance, particularly with Convolutional Neural Networks. The experiments performed on actual cyberbullying data showed a major advantage of this approach to all previous methods, including the best performing method so far based on Brute Force Search algorithm.

The later research in cyberbullying detection has mainly concentrated on using recurrent neural networks and pretrained language models with plain tokens to train embeddings [42, 43, 44, 45]. The exceptions being Balakrishnan et al. [46] and Rosa et al. [47], who used psychological features, like personalities, sentiments and emotions to improve automatic cyberbullying detection. However, these were done using simple models. Using linguistic preprocessing and linguistic embeddings to improve classifier performance has not been studied further even though its potential was confirmed earlier [16].

Vidgen and Derczynski [6] examined over sixty hate speech datasets in 2020. They provided insights into the contents of the datasets, their annotation and the formulation of the associated tasks. They also announced hatespeechdata.com [1], a

[1] `http://hatespeechdata.com`

repository for online abusive content training datasets, in order to make quality data more accessible. Lastly, they provided outlines on best practices for the creation of datasets for online abuse detection.

Also, the popularization of multilingual neural models has made it possible to train models for low-resource languages by utilizing transfer learning. As with cross-lingual transfer, the performance on low-resource languages can be improved by leveraging knowledge from other higher resource languages [48].

Ranasinghe et al. [27, 28] showed the effectiveness of cross-lingual transfer in offensive language identification in Hindi, Spanish, Danish, Greek and Bengali. Their work showed that multilingual transformer models like mBERT and XLM-R can use the knowledge gained from higher resource languages to gain an improved performance on a low-resource target. Also, the models scored comparatively high without any data from the target language, demonstrating the power of cross-lingual pre-training.

Similar results were obtained by Bigoulaeva et al. [29] with English and German. They also discovered that using unlabeled samples from the target language can be used to increase performance. Finally, Gaikwad et al. [30] noticed that transfer learning from Hindi outperformed other languages when classifying entries in Marathi, suggesting a relation between cross-lingual transfer performance and language similarity.

2.2 Model Training Efficiency and Feature Density

2.2.1 Model Performance Estimation

The first research that studied automatic estimation of ML classifier performance was based on extrapolating results from a smaller dataset to simulate the performance of a larger dataset. Basavanhally et al. [49] applied this method to estimating classifier performance in the field of computer aided medical diagnostics, where the available data is in many cases limited in quantity. Their research showed that using a repeated random sampling method on small datasets to estimate the performance on a larger set often had high error rates and should not be generalized as holding true when higher quantities of data become available. Later,

a cross-validation sampling strategy was added, which improved the method and resulted in lower error rates [50].

However, they only used three different classifiers, two of which are widely regarded as low-performance baseline classifiers (Naive Bayes, k-Nearest Neighbor) and Support Vector Machines. Their results suggested that the ranking of the classifiers would not change as the dataset size increases but considering the classifiers used, the results seem very predictable and match the average rankings of these classifiers [51]. Also, they did not consider any deep learning -based approaches. This increases the questionability of the results.

In the field of NLP, Johnson et al.[52] used the fastText classifier [53] and applied the extrapolation method to document classification. They discovered that biased power law model with binomial weights could be used as a good baseline extrapolation model for NLP tasks. As the authors suggested, the method needs to be studied further using different classifiers and extrapolation models. Even though the extrapolation method would aid in estimating the classifier performance, at least some training is still required.

Gama et al. [54] proposed another method, in which classifier performance could be estimated by training a regression model based on meta-level characteristics of a dataset. The characteristics used included simple measures like number of instances and number of attributes, statistical measures like standard deviation ratio and various information based measures like class entropy. These measures are defined in the STATLOG project [55].

This meta-learning based method was adopted and further developed by introducing the Landmarking process [56], which uses the learners themselves to characterize the datasets. This involves using computationally non-demanding classifiers, such as Naive Bayes (NB), to gain valuable insights into datasets. The system outperformed the previous characterization approach and had modest results in the ranking of learners.

Later, Blachnik et al. [57] enhanced the Landmarking method. He proposed that the information from instance selection methods could be used as landmarks. These instance selection methods are usually used for cleaning the dataset, reducing its size by removing redundant information. They found out that it is possible to use the relation between the initial and reduced datasets as a landmark in order to

predict classifier performance with lower error rates.

A more recent method focused on generalizing meta information of the dataset to quantify an overall dataset complexity measure and use it to find correlations between this measure and classifier performance without any training [17]. I describe this research in detail in the following Section 2.2.2.

The problem with a meta-learning based approach is that we would need a lot of datasets and features in order to produce a proper model to estimate classifier performance. The example meta dataset used by Gama et al. [54] was considerably small, consisting of around 20 datasets, which makes its statistical reliability very weak as the error rates get high [58, 59, 60]. The later experiments [56] used around 200 datasets, which is still very underwhelming considering the dataset sizes used today, leading the results with meta-learning being modest at best. Similar problems apply to extrapolation as the subset will get too small to represent the whole dataset and the algorithm is unable to thoroughly capture the characteristics and variations associated with each class. This particularly hurts models like CNNs that rely on larger quantities of data [60]. Also, extrapolation loses vocabulary in the field of NLP, which hinders the applicability of results [61].

In this research, instead of concentrating on training an additional classifier on meta information extracted from the dataset or performance simulation and extrapolation, I directly focused on feature engineering and the relation between the available feature space and classifier performance. This is a novel method that can be utilized together with the existing methods to better estimate the performance of different classifiers.

This research takes a somewhat similar approach as Rahane et al. [17] did for image recognition by directly using a characteristic of the dataset in order to estimate complexity. In the case of this research, this characteristic is Feature Density, based on the notation of Lexical Density [22] that has been used to estimate language complexity. This characteristic was chosen as in previous research [16] where it was shown that there could be a correlation between Feature Density and classifier performance. The research also studies the possibility of preserving linguistic information as parts of used word embeddings in order to improve performance.

2.2.2 Dataset Complexity Estimation

In the field of image recognition, Rahane et al. [17] proposed a method to estimate dataset complexity using various entropy-based information measures on image grayscale values. All of the used measures correlated with the known performance metrics of the applied datasets. They managed to properly rank their datasets using these metrics, which showed that there is a potential in ranking classifiers using these metrics. In the future, a similar method based on word similarity could be applied in NLP.

It has been widely known in linguistics that various languages have different complexity. Therefore language complexity in general is a topic that has been studied throughout the years in linguistics [62]. Especially, attempts to estimate a general lexical complexity of a language [18, 19, 20, 21] or a general syntactic complexity of a language [22, 23, 24, 25] are well documented within the fields of language education. It has been used especially in second language learning, to estimate the level of difficulty to certain first language speakers to learn a specific second language (e.g., how hard it is for native speakers of English to learn Polish or Japanese, which have completely different syntactic complexity). However, to the best of my knowledge, there has not been relevant research in applying this to locally estimate linguistic dataset complexity. One of the measures used for estimating linguistic complexity in general has been Lexical Density (LD) [22]. It is a score representing an estimated measure of content per lexical units for a given corpus, calculated as the number of all distinct words, or vocabulary size, divided by the number of all words in the corpus: $LD = V/N$. This notation means that the corpus is the more complex (or “dense"), the more distinct words it contains. In this research I take a look at an extension of Lexical Density, namely, Feature Density [16], and study its uses in estimating the complexity of different natural language datasets, depending on the kinds of feature sets were used to represent the dataset.

Also, information criterion based methods, like Akaike Information Criterion (AIC) [63] and Bayesian Information Criterion (BIC) [64], have been proposed for model selection since the same time as lexical density was introduced. However using these metrics requires training all of the models which is exactly what I am trying to avoid as feature density can be calculated from the dataset directly

without any training.

2.2.3 Feature Density

The concept of Feature Density (FD) was introduced by Ptaszynski et al. [16] based on the notion of LD [22] from linguistics. The score is called Feature Density as it includes not only lexemes (words/tokens) but also other features, like parts-of-speech, dependency information, or any other applied feature set, in addition or instead of words/tokens

$$FD = \frac{V_x}{N_x}, \tag{2.1}$$

where V_x is the number of distinct features in the corpus and N_x is the total number of features in the corpus. By feature I mean preprocessed (lemmatized, added POS, dependency information etc.) words/tokens. In this research I applied this notation to estimate dataset complexity, by defining it as the complexity of the language dataset itself.

I hypothesize that there could be a correlation between FD and the classifier performance. If FD has a clear positive or negative correlation with classifier performance, it could be useful in comparatively estimating the performance of various classifiers within a dataset. Using the Japanese CB dataset, Ptaszynski et al. [16] showed that CNNs work well with higher FD while other classifiers' scores were usually higher with the lower FD datasets. This shows that it might be possible to improve the performance of CNNs by increasing the FD of the dataset in question, whereas other classifiers could score higher if FD was reduced [16].

I also hypothesize that FD could be useful in comparing different datasets in terms of comparative complexity, as the measure has already been used to estimate linguistic complexity in general [22]. This would be useful in optimizing classifier performance or even estimating it prior to experimentation. In this research, the concept of Feature Density is applied to multiple languages in the field of cyberbullying as well as Sentiment Analysis.

2.2.4 Linguistically-backed Preprocessing

In almost all cases, word embeddings are learned only from pure tokens (words) or in some cases, lemmas (unconjugated forms of words). This also applies to the recently popularized pre-trained language models like BERT [34]. Although word embeddings have been used to help predict parts of speech [65], named entity recognition [66], or dependency parsing [67], utilizing such linguistic information in the process of training the embeddings themselves have not yet been researched extensively, with only a few related introductory studies being proposed so far [68, 69, 70].

There are only a handful of studies related to the usage of linguistic information in training word embeddings. In 2014, Levy and Goldberg [68] modified the Skip-Gram model used by Word2Vec [71] to use dependency structures as contexts while training the word vectors instead of using only a fixed window of surrounding words. They noticed that their dependency-based embeddings were noticeably different from the ones trained with words as contexts as they seemed to be more functional instead of topical. Their method was later evaluated by Komninos and Manandhar [69] and MacAvaney and Zeldes [72]. They acknowledged that dependency-based embeddings outperform the use of linear context in many tasks, especially question classification and semantic relation identification.

In 2017 Ptaszynski et al.[16] proposed a method of adding linguistic information like POS, NER and dependency structures, to the creation of bag-of-words (BoW) models. This showed an improvement to ordinary BoWs when using a Convolutional Neural Network (CNN) model. Their study also hinted that increasing (or decreasing) the density of the used feature set could result in an increased performance. In 2019, Cottorell and Schutze [70] proposed a method of keeping morphological information, like POS, case, gender etc. to encode the words' morphology. They showed that it is possible to encode such information better than Word2Vec by using a modified Log-Bilinear model [73].

To conduct further investigation on the potential of capturing deeper relations between lexical items and structures and to filter out redundant information, I propose to preserve the morphological, syntactic and other types of linguistic information by combining them with the pure tokens or lemmas. This means, for example, including parts-of-speech or dependency information within the used

lexical features. The word embeddings can then be trained using these features instead of just plain tokens. It is also possible to later apply this method to the pre-training of huge language models and possibly enhance their performance.

2.3 Linguistic Similarity and Cross-Lingual Transfer

2.3.1 Measuring Linguistic Similarity

Already in 2006, the relation between the difficulty of language learning and the similarity of languages in general was discussed in a book by Ringbom [74]. The Finnish language scene was presented as an example in order to demonstrate the importance of cross-linguistic similarity in foreign language learning [75]. In short, he showed that Finnish-speaking Finns have a harder time learning English than Swedish-speaking Finns. The reason behind this being the closer relation between Swedish and English languages, giving an advantage to Swedish speakers when it comes to transferring the existing linguistic knowledge.

Cottorell et al. [76] showed that not every language is equally difficult to model. It was also shown by them that there is a correlation between the morphological richness of a language and the performance of the model. This means that the more complex the language is, the more difficult it becomes to model. This is hinting that more simple languages might not work so well when used as cross-lingual transfer sources for languages of higher complexity. This also implies that the direct relatedness (for example, language family) of languages should not be the only criteria in deciding the cross-lingual transfer source language as other features of the languages should also be thoroughly considered in order to find the most optimal transfer language.

There has been some research in attempting to quantify a linguistic similarity metric from different linguistic features. However, these metrics mostly commonly rely only on one or just a few different linguistic features. For example, by comparing the consonants contained in a predefined set of words while taking into account the order in which these consonants appear in the words, one can calculate a genetic proximity score between two languages. This is implemented as the eLinguistics [77]

similarity metric. The metric makes it possible to get information about the direct relatedness of the compared languages. However, it was found out that that once the used languages start to become more and more distant, accidental similarities in consonants are introduced and there is a significant increase in the error rate. Even though the metric is easy to calculate, it completely ignores all other kinds of linguistic features, for example, semantic, syntactic, or morphological.

Another method to calculate a similarity metric is to take a look at the vocabularies of two languages and concentrate on their similarity. EzGlot [78] uses lexical similarity as its basis for computation. The metric uses lexical similarity between the two compared languages while at the same time taking into account the amount of words the two languages are sharing with other languages. This allows for the calculation of similarity between the two languages in relation to the similarity with every other language.

Aggarwal et al. [79] proposed a linguistic similarity metric that utilizes multiple aspects of languages. Their metric, called STL, is based on Semantic, Terminological (lexical) and Linguistic (syntactic) similarity of languages. The method outperformed previous similarity metrics that concentrated only on one of the previously mentioned aspects [80, 81]. They noticed that the terminological measures showed a much higher contribution when compared to the other two features. However, in order to use the metric, the structure of the used vocabulary dataset needs to be in the form of an ontology. Due to this fact and because of the lack of available languages for the used dataset, it was impossible to use the metric as a part of this research.

The lang2vec developed by Littell et al. [82] is a database that represents languages as typological, phylogenetic, and geographical vectors, which are derived from a number of different linguistic resources, for example, WALS [83], PHOIBLE [84], Ethnologue [85], and Glottolog [86]. Each of these utilize multiple different features, making them more robust than the EzGlot or eLinguistics metrics. However, the heterogeneous nature of the method brings up many questions. For example, it is unknown how features are selected from different sources and how they are weighted. Also, using geographical information seems questionable as it has been proven to be a poor predictor of language similarity [87].

2.3.2 Transfer Language Selection

Selecting the optimal language for cross-lingual transfer remains mostly an unanswered question. Most of the time, the decision of which language to use as the transfer source comes up to the practitioner's consideration. This is usually done experimentally or simply by relying on intuition [88]. For example, in order to get a more successful transfer, Cottorell and Heigold [31] focused on using languages from the same language family as the cross-lingual transfer target. However, even though the languages are part of the same language family, two languages could be very distant for example when looking at the complexity of grammar, which means that it does not guarantee them sharing the same linguistic features [32].

A common way for choosing the transfer language is to simply default to English. The reason being that it is the de-facto highest resource language available for most NLP tasks [89]. This is also the case with popular multilingual benchmarks like XTREME [90] and XGLUE [91]. Although, recently benchmarks like XTREME-R [92] have started to include cross-lingual training sets. Furthermore, in a survey of 157 cross-lingual learning papers by Pikuliak et al. [93] they found out that English was used in 149 papers, followed by German with 82 papers. Additionally, it has been shown that other languages than English, for example, German and Russian tend to work better as transfer sources [94].

Duong et al. [95] found out that choosing the transfer language based on language family is not optimal for many languages. For example, their experiments showed that the best source language for both Finnish and German is Czech, even though being from a different language family than the targets. They concluded that apparently, the best source language for cross-lingual transfer is not predictable from language family information. Instead, they proposed two methods for transfer language selection. The first being based on the Jensen-Shannon divergence between the distributions of parts-of-speech n-grams on a pair of languages. The second method was based on the word-order information feature in WALS. Both of these methods showed improvements over choosing English or a language from the same family as the target. They also experimented with using multiple source languages, which further improved the performance.

It has been shown [96, 97, 98] that transferring from many high-resource languages at the same time can yield higher results compared to selecting only a

single language as the transfer source. However, these methods do not consider the actual relation between the source and the target languages and the amount of contribution of each of the languages to the total score. Also, Nooralahzadeh et al. [99] discovered that certain morphosyntactic features shared between languages tend to give a boost to cross-lingual transfer performance.

Lin et al. [100] developed a ranking method for possible transfer language candidates using the lang2vec metrics [82] together with dataset dependent features like word overlap and type-token ratio. they discovered that using both the dataset independent linguistic features and database dependent features to train the ranking model yields the best results. However, as their method requires training of the ranking model, it is dependent on the tasks and datasets used for training and is not usable out of the box for other applications.

In another study [101] it was shown that the transfer performance with English as the source correlates with the linguistic similarity metrics of lang2vec [82], meaning that target languages more similar to English yielded higher scores. They found out that similarity of syntactic structures especially play an important role in selecting the source language for tasks like parts-of-speech tagging (POS), named entity recognition (NER) and dependency parsing (DEP). They also discovered that the fine-tuning corpus size of the target language also makes a difference considering the cross-lingual transfer performance, especially for higher level tasks like question answering. However, their research concentrated only on using English as the source language and the capabilities of other languages as the transfer source were left completely unexplored.

Martinez et al. [102] found out that differences in language morphology in cross-lingual transfer generally lead to a higher loss than when transferring between languages with the same morphological typology. Furthermore, they showed that parts-of-speech tagging tends to be more sensitive towards changes in morphological typology compared to sentiment analysis, which seems to be more sensitive to variables related to the fine-tuning data and the transfer performance being generally harder to predict.

In their research, Gaikwad et al. [30] discovered that there could be a relation between cross-lingual transfer performance and language similarity. They classified entries in the Marathi language using multiple languages, specifically Bengali,

Greek, English, Turkish and Hindi as cross-lingual transfer sources. Their results showed that the closest language of these to Marathi, Hindi, also had the highest performance. This hints that a solution to the problem of cross-lingual transfer language selection could be found with the aid of linguistic similarity, at least for offensive language detection.

2.3.3 The World Atlas of Language Structures

The World Atlas of Language Structures (WALS) project [83] consists of a database that catalogs phonological, word semantic and grammatical knowledge for 2,662 languages with almost two hundred different linguistic features from multiple domains. Using a linguistic similarity measure quantified from the WALS database into would allow a more robust method to measure similarity and would aid capturing all aspects of the languages instead of relying only on a single or a handful of linguistic features. Concentrating purely on using WALS to create a similarity metric would also preserve homogeneity and allow a more explainable and controllable implementation. I hypothesize that this metric would be more robust compared to the other metrics as it is based on multiple kinds of linguistic features. Instead of concentrating on lang2vec, I decided to develop the previously introduced novel metric based on the World Atlas of Language Structures, which also contains a variety of phonological, grammatical and lexical features.

However, many of the linguistic features are missing for of the available languages. For example, one of the most extensively documented language, English, has about 150 features documented in the database. This amount rapidly decreases for languages studied less. Taking Danish as an example, it only 58 features documented [2]. Considering every language and all of the features, this adds up to over 58,000 data points in total in the WALS database. This means the whole database is only approximately 12% populated, meaning a vast majority of the information is missing. Also many major and widely studied languages are missing many features. For example, 25% of all of the features are missing for English. These missing values and the sparsity of the data is the main point of concern when

[2]Some even less studied languages have an even smaller number of features documented, e.g. Chuj language, spoken in Guatemala, has only 29, while the Indonesian Kutai language has only a single feature documented.

quantifying the WALS database into a linguistic similarity metric as using lesser known and not so widely studied languages means having less common features among them.

Chapter 3

Classifier Performance Estimation with Feature Density

In this Chapter, I apply the concept of Feature Density and explore its potential in comparatively estimating the efficiency of various ML classifiers prior to any training. This is done in order to hasten the development of cyberbullying detection models, by estimating classifier performance prior to the training process in order to reduce the number of required experiment iterations. Additionally I evaluate the usefulness of numerous linguistically-backed feature preprocessing approaches to create more effective models for detecting bullying.

I calculate FD for all applied datasets and preprocessing methods. I hypothesize that FD would correlate with the classification scores. If FD has a clear positive or negative correlation with classifier performance, it could be useful in comparatively estimating the performance of various classifiers within a dataset. In practice, I calculated Pearson's correlation coefficient (ρ-value) between dataset generalization (FD) and classifier results (macro F-scores). Using the Japanese CB dataset, Ptaszynski et al. [16] showed that CNNs work well with higher FD while other classifiers' scores were usually higher with the lower FD datasets. This shows that it might be possible to improve the performance of CNNs by increasing the FD of the dataset in question, whereas other classifiers could score higher if FD was reduced [16].

I also compare different datasets in terms of comparative complexity. Using Feature Density and the experiment results, I study the usefulness of estimating

Table 3.1: Statistics of the applied cyberbullying datasets.

	English	**Japanese**	**Polish**
Number of samples	12,772	2,998	11,041
Number of CB samples	913	1,490	985
Number of non-CB samples	11,859	1,508	10,056
Number of all tokens	308,939	39,283	142,811
Number of distinct tokens	25,106	6,947	27,444
Avg. length (chars) of a sample	125.5	35.2	95.5
Avg. length (words) of a sample	28.7	13.3	14.3
Avg. length (chars) of a CB sample	115.4	33.5	105.1
Avg. length (words) of a CB sample	26.8	12.6	15.0
Avg. length (chars) of a non-CB sample	126.3	37.0	94.5
Avg. length (words) of a non-CB sample	26.8	14.1	14.2

dataset complexity in optimizing classifier performance or even estimating it prior to experimentation. I apply the concept of Feature Denxity to multiple languages in the field of Cyberbullying as well as verify the results on a Sentiment Analysis task.

3.1 Applied Datasets

To achieve a thorough and validated analysis, I applied the proposed methods to multiple languages, namely, Japanese, English and Polish. This also allowed to study the effect of linguistic and cultural differences in classifier performance when the same classification methods are used for different languages. The research uses a total of four datasets, three from the field of automatic cyberbullying detection and one verification set from sentiment analysis. Key statistics of the applied cyberbullying datasets are shown in Table 3.1.

3.1.1 English Cyberbullying Dataset

The first dataset for the experiments was the Kaggle Formspring Dataset for Cyberbullying Detection [103]. There was one major problem with the original

dataset however, as the original annotations for the data were carried out by untrained laypeople. It has been shown before that the annotations for topics like online harassment and cyberbullying should be done by experts [5] as the abusive content could be hidden for example in supposed humor, sarcasm or dank memes. Therefore, the dataset was re-annotated with the help of experts with sufficient psychological background to assure high quality annotations [104]. In the research I applied the re-annotated version for more accurate results.

Table 3.1 reports some key statistics of the improved annotation of the dataset. The dataset contains approximately 300 thousand of tokens. There was no visible difference in length between the posted questions and answers, both being approximately 12 words long on average. On the contrary, the harmful (CB) entries were usually slightly but insignificantly shorter compared to the non-harmful (non-CB) samples (approx. 23 vs. 25 words). The amount of harmful samples was also substantially smaller compared to the amount of non-harmful samples, around 7% of the whole dataset, which is approximately the same as the real-life amount of profanity encountered on SNS [5].

3.1.2 Japanese Cyberbullying Dataset

The Japanese dataset I used for experiments was originally created by Ptaszynski et al. [9], and also widely used by others [38, 41, 11, 15, 16]. It contains 1,490 harmful and 1,508 non-harmful entries in Japanese collected from unofficial school websites and fora. The original data was provided by the Human Rights Research Institute Against All Forms for Discrimination and Racism in Mie Prefecture, Japan. The entries were collected and labeled by Internet Patrol members (expert annotators) with the help of the government supplied manual [14]. The instructions given by the manual are briefly described below.

The definition given by MEXT suggests that cyberbullying occurs when a person is directly offended on the Internet. This includes publication of the person's identity, personal information and other aspects of privacy. Thus, as the first distinguishable features for cyberbullying, MEXT identifies private names (also initials and nicknames), names of organisations and affiliations and private information (address, phone numbers, personal information disclosure, etc.)

In addition, cyberbullying literature reveals vulgarities as one of the most distin-

guishing characteristics of cyberbullying [1, 105]. Also according to MEXT, vulgar language and cyberbullying can be distinguished from each other as cyberbullying conveys offenses against real individuals. In the prepared dataset, all entries containing at least one of the above characteristics is listed as harmful.

3.1.3 Polish Cyberbullying Dataset

The Polish dataset originates from PolEval workshop from 2019 [106], collected from Twitter discussions. As feature selection and feature engineering have been proven to be integral parts of cyberbullying detection [16, 107], the tweets are provided as such, without additional preprocessing to allow researchers apply their own preprocessing methods. The only preprocessing applied to the dataset was done only to mask private information, such as personal information of individuals (Twitter users).

The dataset contains 11,041 entries in total, with 10,041 included in the training set and 1000 in the test set. The dataset was initially annotated by laypeople, but was later corrected by an expert in the case of disagreements. The laypeople agreed on majority of the annotations at 91.38%. The number seems very high, but it is mostly due to the fact that the annotators mostly agreed upon non-harmful tweets, which take up most of the dataset at 89.76%. When considering the harmful class, the annotators only agreed upon 1.62% of the entries. Moreover, some of the fully-agreed tweets needed to be corrected to the opposite class in the end by the expert annotator, which shows that using laypeople does not provide accurate enough annotations in the field of cyberbullying. It could be said that layperson annotators can tell with a decent level of confidence that an entry is not harmful (even if it contains some vulgar words), and they can spot, to some extent, if the entry is somehow harmful. Though in most cases they are unable to provide a reasoning for their choice. This provides further proof that for specific problems such as cyberbullying, an expert annotation is required [5]. Comparing the training and test sets, it can be noted that the latter contained a slightly higher ratio of harmful tweets (8.48% for training set vs. 13.40% for test set), which might end up showing in the classifier evaluations.

There was also a reasonably high number of retweets that have slipped into both the data processing phase and the annotation (709 or 6.42%). All of these

tweets were not official retweets made using the retweet function, but tweet quotes beginning with a brief "RT" statement, which differentiates them from normal replies and comments. This needs to be taken into account in the future.

3.1.4 Verification Dataset: Yelp User Reviews Sentiment Dataset

As a dataset for verification of the claimes posed in this study, I applied a subset of Yelp's user reviews data. It contains business reviews about restaurants, shops, etc. from North American metropolises with a rating from one to five stars along with other information. The dataset was originally assembled for Yelp's dataset challenge in order to promote innovation.

The ratings were binarized for the experiment by assigning one star reviews to negative class and five star reviews to positive class. Other reviews were discarded. Also, reviews containing less than three words or more than two standard deviations from the mean were filtered out to avoid over-lengthy review samples differing too much with average sample length in other datasets. I took a random subset of 250,000 positive and 250,000 negative reviews. This way I could study the influence of the change in size of a dataset by a roughly over one magnitude but no more than two magnitudes. With these constraints, I had a large binary dataset of a different topic that is also much simpler than the cyberbullying datasets. Although the whole dataset was much larger and had multiple classes, difference of more than two magnitudes between other datasets and inclusion of more classes might introduce uncontrollable and untraceable variations in dataset statistics influencing the measurements. Therefore, applying these limitations would allow to study the proposed methods with sufficient control. The subset was split into training and test datasets containing 80% and 20% of the data respectively with even number of positive and negative samples in each part.

I also chose the dataset deliberately from a different field in order to verify the general potential of the performance of the proposed method and its universal applicability. The dataset was also considerably larger, more specifically, almost fifty times that of the Polish or English datasets. This made it possible to study how the proposed methods perform with a larger amount of data. The dataset was

binarized to provide comparability with the performance metrics between all of the datasets.

3.2 Proposed Methods

3.2.1 Preprocessing and Feature Density

In order to train the linguistically-backed embeddings, I first preprocessed the dataset in various ways, similarly to previous research [16]. This was done for three reasons. Firstly, to see how traditional classifiers managed the data from similar domain (cyberbullying), but in different languages. Secondly, to later verify the correlation between the classification results and Feature Density (FD) [16]. Finally, to verify the performance of various versions of the proposed linguistically-backed embeddings. Also, because I was trying to make the proposed method entirely systemic and automated, I did not focus on any hand-made features, such as offensive word lexicons, etc., used in previous research [9]. The preprocessing was done using spaCy NLP toolkit [108]. After assembling combinations from the listed preprocessing types, I ended up with a total of 68 possible preprocessing methods for the experiments. All types of preprocessing I applied to generate preprocessing type combinations were listed below. The FDs for all preprocessing types used in this research are shown in Tables 3.2, 3.3, 3.4 and 3.5.

- **Tokenization:** includes words, punctuation marks, etc. separated by spaces (later: TOK).
- **Lemmatization:** like the above but with generic (dictionary) forms of words ("lemmas") (later: LEM).
- **Parts of speech (separate):** parts of speech information is added in the form of separate features (later: POSS).
- **Parts of speech (combined):** parts of speech information is merged with other applied features (later: POS).
- **Named Entity Recognition (without replacement):** information on what named entities (private name of a person, organization, numericals, etc.)

appear in the sentence are added to the applied word (later: NER).

- **Named Entity Recognition (with replacement):** same as above but information replaces the applied word (later: NERR).
- **Dependency structure:** noun- and verb-phrases with syntactic relations between them (later: DEP).
- **Chunking:** like above but without dependency relations ("chunks", later: CHNK).
- **Stopword filtering:** redundant words are filtered out using spaCy's stopword lists (later: STOP)
- **Filtering of non-alphabetics:** non-alphabetic characters are filtered out for English and Polish. For Japanese, *kanji* and *kana* characters are also retained (later: ALPHA)

3.2.2 Feature Extraction

For classifiers other than those based on neural networks, I generated a Bag-of-Features language model was from each of the 68 processed dataset versions, producing a separate model for each of the preprocessing types (Bag-of-Words, Bag-of-Lemmas, Bag-of-POS, etc.). This was done for all of the four datasets. The language models generated from the entries of the datasets were used later in the input layer of classification. I also applied a traditional weight calculation scheme, namely term frequency with inverse document frequency $tf * idf$, where term frequency $tf(t, d)$ refers to raw frequency (number of times a term t (word, token) occurs in a document d), and inverse document frequency $idf(t, D)$ is the logarithm of the total number of documents $|D|$ in the corpus divided by the number of documents containing the term n_t. Finally, $tf * idf$ refers to term frequency multiplied by inverse document frequency as in equation3.1.

$$idf(t, D) = log(\frac{|D|}{n_t}) \tag{3.1}$$

Table 3.2: English Cyberbullying Dataset: Feature Density of preprocessing types.

Preprocessing type	Uniq.1grams	All1grams	FD
POS	18	357616	.0001
POSALPHA	18	357616	.0001
POSSTOP	18	194606	.0001
POSSTOPALPHA	17	129076	.0001
LEMPOSSALPHA	17875	579664	.0308
LEMPOSS	21238	660653	.0321
TOKPOSSALPHA	21737	579624	.0375
TOKPOSS	25122	660612	.038
LEMNERALPHA	14815	289868	.0511
LEMNERR	17327	309124	.0561
CHNKNERRALPHA	12293	215096	.0572
LEMNERRALPHA	17877	305481	.0585
CHNKNERALPHA	14007	228146	.0614
LEMALPHA	17860	289868	.0616
LEMPOSSSTOP	20948	334870	.0626
TOKNERRALPHA	18595	289828	.0642
CHNKALPHA	13991	215096	.065
LEMNER	21239	325173	.0653
LEMPOSSSTOPALPHA	17554	258103	.068
TOKNERR	21119	309084	.0683
LEM	21222	308434	.0688
TOKNERALPHA	21737	305441	.0712
TOKPOSSSTOP	24472	334869	.0731
LEMPOS	26232	357657	.0733
TOKALPHA	21722	289828	.0749
LEMPOSALPHA	22206	289868	.0766
TOKNER	25121	325132	.0773
TOK	25106	308393	.0814
TOKPOSSSTOPALPHA	21037	258103	.0815
TOKPOS	31121	357616	.087
TOKPOSALPHA	27013	289828	.0932
LEMNERRSTOPALPHA	14509	129076	.1124
LEMNERRSTOP	17047	146549	.1163
LEMNERSTOPALPHA	17557	142289	.1234
CHNKNERR	33025	262529	.1258
LEMNERRSTOPALPHA	20950	160269	.1307
LEMPOSSTOP	25669	194674	.1319
LEMSTOPALPHA	17540	129076	.1359
TOKNERRSTOPALPHA	17911	129076	.1387
CHNKNER	38044	272581	.1396
TOKNERRSTOP	20480	146549	.1397
LEMSTOP	20933	145866	.1435
CHNKNERSTOPALPHA	13356	92782	.144
CHNKNERRSTOPALPHA	11656	80896	.1441
CHNK	38029	261990	.1452
TOKNERSTOPALPHA	21037	142289	.1478
TOKNERSTOP	24471	160268	.1527
TOKPOSSTOP	30040	194673	.1543
TOKSTOPALPHA	21022	129076	.1629
CHNKSTOPALPHA	13340	80896	.1649
LEMPOSSTOPALPHA	21626	129076	.1675
TOKSTOP	24456	145865	.1677
TOKPOSSTOPALPHA	25925	129076	.2009
CHNKNERRSTOP	32452	126357	.2568
CHNKNERSTOP	37462	135357	.2768
CHNKSTOP	37447	125824	.2976
DEPNERALPHA	95404	240302	.397
DEPNERRALPHA	94928	215096	.4413
DEPALPHA	95386	215096	.4435
DEPNER	143197	321835	.4449
DEPNERSTOPALPHA	47159	104940	.4494
DEPNERR	141479	308704	.4583
DEP	143179	308704	.4638
DEPNERSTOP	94539	184130	.5134
DEPNERRSTOP	92730	172086	.5389
DEPSTOP	94521	172086	.5493
DEPNERRSTOPALPHA	46552	80896	.5755
DEPSTOPALPHA	47141	80896	.5827

Table 3.3: Japanese Cyberbullying Dataset: Feature Density of preprocessing types.

Preprocessing type	Uniq.1grams	All1grams	FD
POS	19	40015	0.0005
POSALPHA	19	40015	0.0005
POSSTOP	19	25777	0.0007
POSSTOPALPHA	16	18444	0.0009
LEMPOSS	6495	78685	0.0825
TOKPOSS	6964	79226	0.0879
LEMPOSSALPHA	6152	65314	0.0942
TOKPOSSALPHA	6579	65314	0.1007
LEMPOSSSTOP	6392	50203	0.1273
LEMNERR	5065	39024	0.1298
TOKPOSSSTOP	6818	50742	0.1344
TOKNERR	5474	39380	0.1390
LEMNER	6607	44046	0.1500
LEMNERRALPHA	4918	32665	0.1506
TOKNER	7078	44509	0.1590
TOKNERRALPHA	5329	32665	0.1631
LEMPOSSSTOPALPHA	6049	36846	0.1642
LEM	6478	38955	0.1663
LEMNERALPHA	6266	36835	0.1701
LEMPOS	6870	40017	0.1717
TOKPOSSSTOPALPHA	6433	36846	0.1746
TOK	6947	39283	0.1768
CHNKNERR	5911	32935	0.1795
TOKNERALPHA	6694	36835	0.1817
TOKPOS	7505	40018	0.1875
LEMALPHA	6138	32665	0.1879
CHNKNERRALPHA	5289	26694	0.1981
LEMPOSALPHA	6520	32665	0.1996
LEMNERRSTOP	4961	24791	0.2001
TOKALPHA	6564	32665	0.2009
TOKNERRSTOP	5328	25143	0.2119
CHNKNER	7797	36061	0.2162
TOKPOSALPHA	7109	32665	0.2176
LEMNERSTOP	6504	29641	0.2194
CHNKNERALPHA	6659	29873	0.2229
TOKNERSTOP	6932	30102	0.2303
CHNK	7672	32877	0.2334
CHNKALPHA	6534	26694	0.2448
LEMSTOP	6375	24722	0.2579
LEMPOSSTOP	6700	25779	0.2599
LEMNERRSTOPALPHA	4814	18444	0.2610
TOKSTOP	6801	25046	0.2715
LEMNERSTOPALPHA	6163	22444	0.2746
TOKPOSSTOP	7237	25780	0.2807
TOKNERRSTOPALPHA	5183	18444	0.2810
CHNKNERRSTOP	5774	20190	0.2860
TOKNERSTOPALPHA	6548	22444	0.2917
LEMSTOPALPHA	6035	18444	0.3272
CHNKNERSTOP	7659	23283	0.3290
LEMPOSSTOPALPHA	6350	18444	0.3443
TOKSTOPALPHA	6418	18444	0.3480
CHNKNERRSTOPALPHA	5152	14003	0.3679
TOKPOSSTOPALPHA	6841	18444	0.3709
CHNKSTOP	7534	20132	0.3742
CHNKNERSTOPALPHA	6521	17132	0.3806
CHNKSTOPALPHA	6396	14003	0.4568
DEPNERSTOPALPHA	12078	19727	0.6123
DEPNERALPHA	21542	32541	0.6620
DEPNERSTOP	16800	23802	0.7058
DEPNER	26264	36581	0.7180
DEPNERR	26089	33357	0.7821
DEP	26139	33357	0.7836
DEPNERRALPHA	21354	26694	0.8000
DEPALPHA	21417	26694	0.8023
DEPNERRSTOP	16619	20611	0.8063
DEPSTOP	16675	20611	0.8090
DEPNERRSTOPALPHA	11884	14003	0.8487
DEPSTOPALPHA	11953	14003	0.8536

Table 3.4: Polish Cyberbullying Dataset: Feature Density of preprocessing types.

Preprocessing type	Uniq.1grams	All1grams	FD
POS	15	157137	0.0001
POSALPHA	15	157137	0.0001
POSSTOP	15	104715	0.0001
POSSTOPALPHA	15	63004	0.0002
LEMPOSS	16403	294706	0.0557
LEMPOSSALPHA	14952	230795	0.0648
LEMPOSSSTOP	16246	189900	0.0856
TOKPOSS	27458	294717	0.0932
LEMNERR	14509	142929	0.1015
LEMPOS	17217	157137	0.1096
LEMNER	16393	147531	0.1111
TOKPOSSALPHA	26067	230795	0.1129
LEMNERRALPHA	13167	115398	0.1141
LEM	16388	142800	0.1148
LEMPOSSSTOPALPHA	14756	125979	0.1171
LEMNERALPHA	14943	121392	0.1231
LEMALPHA	14938	115398	0.1294
LEMPOSALPHA	15479	115398	0.1341
TOKPOSSSTOP	26839	189911	0.1413
LEMNERRSTOP	14350	90569	0.1584
LEMPOSSTOP	16923	104722	0.1616
LEMNERSTOP	16236	95020	0.1709
TOKNERR	24922	142940	0.1744
CHNKNERR	24922	142940	0.1744
LEMSTOP	16231	90441	0.1795
TOKPOS	28521	157137	0.1815
TOKNER	27450	147542	0.1860
CHNKNER	27450	147542	0.1860
TOK	27444	142811	0.1922
CHNK	27444	142811	0.1922
TOKPOSSSTOPALPHA	25379	125979	0.2015
TOKNERRALPHA	23630	115398	0.2048
CHNKNERRALPHA	23630	115398	0.2048
LEMNERRSTOPALPHA	12972	63004	0.2059
CHNKNERALPHA	26060	121997	0.2136
LEMNERSTOPALPHA	14747	68845	0.2142
TOKNERALPHA	26060	121392	0.2147
TOKALPHA	26054	115398	0.2258
CHNKALPHA	26054	115398	0.2258
TOKPOSALPHA	26784	115398	0.2321
LEMSTOPALPHA	14742	63004	0.2340
LEMPOSSTOPALPHA	15173	63004	0.2408
TOKPOSSTOP	27683	104722	0.2643
TOKNERRSTOP	24303	90580	0.2683
CHNKNERRSTOP	24303	90580	0.2683
TOKNERSTOP	26831	95031	0.2823
CHNKNERSTOP	26831	95031	0.2823
TOKSTOP	26825	90452	0.2966
CHNKSTOP	26825	90452	0.2966
TOKNERRSTOPALPHA	22946	63004	0.3642
CHNKNERRSTOPALPHA	22946	63004	0.3642
CHNKNERSTOPALPHA	25372	69449	0.3653
TOKNERSTOPALPHA	25372	68845	0.3685
TOKSTOPALPHA	25366	63004	0.4026
CHNKSTOPALPHA	25366	63004	0.4026
TOKPOSSTOPALPHA	25930	63004	0.4116
DEPNERSTOP	68279	111161	0.6142
DEPNER	102460	163736	0.6258
DEPNERRSTOP	67378	104715	0.6434
DEPNERR	101554	157137	0.6463
DEPNERSTOPALPHA	51860	80173	0.6469
DEPNERALPHA	86044	132723	0.6483
DEPSTOP	68273	104715	0.6520
DEP	102454	157137	0.6520
DEPNERRALPHA	85138	115398	0.7378
DEPALPHA	86038	115398	0.7456
DEPNERRSTOPALPHA	50959	63004	0.8088
DEPSTOPALPHA	51854	63004	0.8230

Table 3.5: Verification Dataset (English Yelp Reviews): Feature Density of preprocessing types.

Preprocessing type	Uniq.1grams	All1grams	FD
POS	18	22123101	0.0000
POSALPHA	18	22123101	0.0000
POSSTOP	17	10976927	0.0000
POSSTOPALPHA	17	7913333	0.0000
LEMPOSSALPHA	108272	37473399	0.0029
LEMPOSS	124750	41602171	0.0030
TOKPOSSALPHA	125796	37473386	0.0034
TOKPOSS	142140	41602328	0.0034
LEMNERALPHA	81750	18736707	0.0044
LEMNERR	92299	19691797	0.0047
TOKNERRALPHA	100174	18736694	0.0053
CHNKNERALPHA	67433	12588242	0.0054
LEMNERRALPHA	108270	19681446	0.0055
TOKNERR	110454	19691940	0.0056
LEMALPHA	108260	18736707	0.0058
LEMNER	124748	20691409	0.0060
CHNKNERALPHA	82630	13295729	0.0062
LEM	124828	19686486	0.0063
TOKNERALPHA	125792	19681433	0.0064
LEMPOSSSTOP	124382	19312556	0.0064
CHNKALPHA	82619	12588242	0.0066
TOKALPHA	125783	18736694	0.0067
LEMPOSSSTOPALPHA	107695	15826663	0.0068
TOKNER	142136	20691566	0.0069
LEMPOS	157276	22126373	0.0071
TOK	142145	19686629	0.0072
LEMPOSALPHA	136041	18736707	0.0073
TOKPOSSSTOP	141540	19312673	0.0073
TOKPOSSSTOPALPHA	124822	15826663	0.0079
TOKPOS	191314	22126347	0.0086
TOKPOSALPHA	169800	18736694	0.0091
LEMNERRSTOPALPHA	81202	7913333	0.0103
LEMNERRSTOP	91950	8548830	0.0108
LEMNERSTOPALPHA	107692	8613858	0.0125
TOKNERRSTOPALPHA	99226	7913333	0.0125
TOKNERSTOP	109868	8548956	0.0129
LEMNERSTOP	124379	9302597	0.0134
LEMSTOPALPHA	107682	7913333	0.0136
LEMPOSSTOP	156259	10980340	0.0142
TOKNERSTOPALPHA	124817	8613858	0.0145
LEMSTOP	124369	8517231	0.0146
TOKNERSTOP	141535	9302714	0.0152
TOKSTOPALPHA	124808	7913333	0.0158
CHNKNERRSTOPALPHA	66524	4096850	0.0162
TOKSTOP	141526	8517348	0.0166
LEMPOSSTOPALPHA	134752	7913333	0.0170
TOKPOSSTOP	189338	10980327	0.0172
CHNKNERSTOPALPHA	81699	4699728	0.0174
CHNKSTOPALPHA	81688	4096850	0.0199
TOKPOSSTOPALPHA	167504	7913333	0.0212
CHNKNERR	764785	15908671	0.0481
CHNKNER	861736	16452253	0.0524
CHNK	861725	15890287	0.0542
CHNKNERRSTOP	764246	7115714	0.1074
CHNKNERSTOP	861185	7556935	0.1140
CHNKSTOP	861174	7097343	0.1213
DEPNERALPHA	2240284	13497593	0.1660
DEPNERRALPHA	2184827	12588242	0.1736
DEPALPHA	2240266	12588242	0.1780
DEPNERR	3758037	18251727	0.2059
DEPNER	3933231	18959225	0.2075
DEP	3933213	18251727	0.2155
DEPNERSTOPALPHA	1300041	4901505	0.2652
DEPNERSTOP	2972316	10059025	0.2955
DEPNERRSTOP	2795319	9456061	0.2956
DEPNERRSTOPALPHA	1241849	4096850	0.3031
DEPSTOP	2972298	9456061	0.3143
DEPSTOPALPHA	1300023	4096850	0.3173

With the Neural Network models, MLP and CNNs, I trained the embeddings as a part of the network using the previously-described preprocessed datasets. Similarly to other classifiers, I trained a separate model for each of the 68 datasets (Word/token Embeddings, Lemmas Embeddings, POS Embeddings, Chunks Embeddings, etc.). The embeddings were fully trained on the datasets themselves as part of the network using using Keras' [109] embedding layer with random initial weights, meaning no pretraining was used.

3.2.3 Classification

In the experiment I applied the following classification algorithms. The assumption was that each classifier presents different correlation with FD, and characteristics of this correlation can be further exploited to minimize the time required for training optimal solutions by choosing a small number of feature sets which usually perform best, or at least eliminating feature sets which always perform below an acceptable performance threshold.

SVM or Support-vector machines [110] are a set of classifiers well established in AI and NLP. They represent data, belonging to specified categories, as points in space (vectors), and find an optimal hyperplane to separate the examples from each category. SVM has been very successful in previous cyberbullying research [9, 8, 16]. In this research, I used two SVM functions, the **linear SVM**, as it had the greatest performance out of all of the SVM kernel functions that were used in previous research [16] and a linear SVM function supported with Stochastic Gradient Descent optimizer (**SGD**).

NaïveBayes (NB) classifier is a supervised learning algorithm applying Bayes' theorem that has a strong (naïve) assumption of independence between pairs of features. It is traditionally used as a baseline in different text classification tasks and is known for working well with smaller datasets. Also, it is considerably fast to train compared to some other popular classifiers e.g. Random Forest.

kNN or the k-Nearest Neighbors classifier takes as input k-closest training samples with assigned classes and classifies the input sample by a majority vote. It is often applied as a baseline alongside Naïve Bayes. The classifier is fast and simple to train but on the contrary, it is very susceptible to outliers and overfitting. In this research, I used k=1 setting in which the input sample is simply assigned to the

class of the first nearest neighbor.

Random Forest (RF) in training phase creates multiple decision trees to output the optimal class (mode of classes) in classification phase [111]. An improvement of RF when comparing to standard decision trees is the ability to correct overfitting to the training set, which is very common in decision trees [112]. In practice, Random Forest starts by taking a random bootstrap sample with replacement from the dataset. It then selects a random subset of features in order to reduce the dimensionality of the sample. Next, an unpruned decision tree is trained on this bootstrap sample. This process is repeated for the desired ensemble size. The predicted value of an unknown instance is obtained by taking a majority vote over the entire ensemble of trees [111].

Logistic Regression (LR) is a statistical model that calculates class probabilities using a logistic function (sigmoid) instead of a straight line (linear regression) or a hyperplane. Logistic regression models are usually fit using Maximum Likelihood Estimation [112]. The model assigns a probability value between [0,1] for each input, which is used to determine the class it belongs. The experiments in this research are conducted using two different solvers, **Newton's method** and **l-bfgs**, which is a quasi-Newton method that uses approximations and memory saving features in order to improve performance with the cost of possible minor convergence problems.

Boosting includes algorithms such as **AdaBoost** [113] and Extreme Gradient Boosting (**XGBoost**) [114], which is a more generalized and optimized version. In boosting, the weak learners, which are usually decision trees, evolve over time as they are trained sequentially to perform better on the residuals of the previous learner. The members cast a weighted vote instead of generating random predictors (RF) and averaging their result [112].

MLP (Multilayer Perceptron) is a type of feed-forward artificial neural network consisting of an input layer, an output layer and one or more hidden layers. In this experiment MLP refers to a neural network using regular dense layers. I applied an MLP implementation with Rectified Linear Units (ReLU) as a neuron activation function [115] and one hidden layer with dropout regularization to reduce overfitting and improve generalization by randomly dropping out some of the hidden neurons during training [115].

CNN or Convolutional Neural Networks are a type of feed-forward artificial neural

network utilizing convolutional and pooling layers. Although originally designed for image recognition, the effectiveness of CNNs has been shown in multiple other tasks, including NLP [116] and sentence classification [117]. I implemented CNNs with Rectified Linear Units (ReLU) as a neuron activation function, and max pooling [118], which applies a max filter to non-overlying sub-parts of the input to reduce dimensionality and as a result, helps to prevent overfitting. I also applied dropout regularization on penultimate layer for the same reason. I applied two versions of CNNs. First, with only one hidden convolutional layer containing 128 units. The second network consisted of two hidden convolutional layers with 128 feature maps each, 4x4 size of patch and 2x2 max-pooling. I used Adaptive Moment Estimation (Adam), a variant of Stochastic Gradient Descent [119] as the optimization function.

3.3 Experiments

3.3.1 Setup

The four preprocessed datasets were additionally preprocessed according to methodology described in Section 3.2.1, which resulted in the creation of 68 separate training sets for each original dataset. The experiment was performed once for every preprocessing type for every dataset. Each of the classifiers (sect. 3.2.3) were tested on all of the versions of the datasets in a 10-fold cross validation procedure for the English and Japanese Cyberbullying datasets and with predetermined train-test set splits for the Polish and verification datasets, to retain the originally proposed method of evaluation for each dataset. This gave an opportunity to evaluate how effective different preprocessing methods were for each classifier and for each language, also with comparison to previous methods. As some of the datasets were not balanced, I oversampled the minority class using Synthetic Minority Over-sampling Technique (SMOTE) [120] in order to balance out the classes. The preprocessing methods represent a wide range of Feature Densities, which can be used to evaluate the correlation with classifier performance. This gave nearly eighteen thousand experiment runs[1] which I use as a basis for the discussions on the results and applicability of FD. The hardware used for the experiments

[1] 2 datasets x 10 fold x 68 preprocessings x 12 classifiers (16,320 runs) + 2 datasets x 1 train-test x 68 preprocessings x 12 classifiers (1,632) = 17,952 experiment runs.

Table 3.6: Runtimes and approximate power usage of the training processes. Non-neural classifiers: Intel i9 7920X@2.90 GHz, 163W. Neural classifiers: Nvidia GTX 1080ti, 250W. Expecting 100% power usage.

	English		Japanese	
Classifier	**Runtime (s)**	**Power usage (Wh)**	**Runtime (s)**	**Power usage (Wh)**
Logistic Regression	321.6	145.61	82.75	37.47
CGD LR	249.74	113.08	76.74	34.75
SGD SVM	176.26	79.81	11.35	5.14
Linear SVM	1543.06	698.67	41.41	18.75
KNN	556.44	251.94	43.22	19.57
Naive Bayes	97.54	44.16	36.36	16.46
Random Forest	3982.49	1803.18	420.19	190.25
AdaBoost	10425.4	4720.39	611.21	276.74
XGBoost	17917.74	8112.76	47093.99	21323.11
MLP	53845.89	37392.98	18235.85	12663.78
CNN1	62361.45	43306.56	21288.6	14783.75
CNN2	62054.46	43093.37	16116.72	11192.17
	Polish		**Yelp**	
Classifier	**Runtime (s)**	**Power usage (Wh)**	**Runtime (s)**	**Power usage (Wh)**
Logistic Regression	56.66	25.65	3429.78	1552.93
CGD LR	45.42	20.56	3592.51	1626.61
SGD SVM	160.06	72.47	2876.84	1302.57
Linear SVM	36.82	16.67	28365.17	12843.12
KNN	56.75	25.69	2687.75	1216.95
Naive Bayes	33.77	15.29	35976.27	16289.26
Random Forest	118.39	53.61	5420.16	2454.13
AdaBoost	136.26	61.7	3429.78	1552.93
XGBoost	2042.64	924.86	3592.51	1626.61
MLP	7095.01	4927.09	121782.6	84571.25
CNN1	9939.68	6902.56	144275.23	100191.13
CNN2	9418.16	6540.39	156572.35	108730.8

included Intel i9 7920X, running at stock 2.90 GHz, for non-neural classifiers and Nvidia GTX 1080ti for neural classifiers. The power consumptions were calculated expecting 100% power usage for the device in question.

3.3.2 Effect of Feature Density

3.3.2.1 English Cyberbullying Dataset

I trained each of the classifiers using the proposed preprocessing methods. The classification results are presented in Table 3.7. As the results for using only parts-of-speech tags, which had the lowest FD by far, were extremely low (close to a coinflip). Thus, I can say that POS tags alone do not contain enough information to successfully classify the entries. I also analyzed the correlation between Feature

Density and classifier F-score, which is shown in Table 3.8.

After excluding the preprocessing methods that only used POS tags, all classifiers, except CNNs have a strong negative correlation with Feature Density. So these classifiers seem to have a weaker performance if a lot of linguistic information is added, and the best results being usually within the range of .05 to .15 of FD depending on the classifier. This range includes 38 of the 68 preprocessing methods (Table 3.2). The sweet spot for performance can be seen from, for example, the highest performing classifier, SVM with SGD optimizer (Figure 3.1), where the maximum classifier performance starts high at around .05 of FD and slowly falls until .14 after which there is a noticeable drop. The performance only falls further as the FD rises. If the weaker feature sets were to be left out, the power savings are approximately 35Wh calculated from Table 3.6 for training the SGD SVM classifier, which is not very much. But the classifier was very power efficient to train to begin.

For CNNs however, there was a very weak positive or no correlation between FD and the classifier performance, with the higher FD datasets performing equally or even slightly better when comparing to the low FD datasets. Taking a look at one layer CNN's performance, which was better than the CNN with two layers, one can see from Figure 3.1 that the maximum performance starts at a moderate level and stays stable throughout the whole range of feature densities. The best results can be found between .05 to .1 and after around .45 FD. From Table 3.2 one can see that around half of the preprocessing types fall into these ranges. One can calculate from Table 3.6 that the power savings for the CNN are approximately 21kWh, which is huge compared to the SVM's power savings.

The results suggest that for non-CNN classifiers there is no need to consider preprocessings with a high FD, such as chunking or dependencies, as they had a considerably lower performance. The performance seems to start falling rapidly at around .15 FD with most of the classifiers. For CNNs, as there was almost no correlation between FD and F-score, I am unable to estimate an ideal range for Feature Density. However, this means that there is a potential in the higher FD preprocessing types, namely, dependencies for CNNs.

The reason for CNNs relatively low performance could be explained by the relatively small size of the dataset, especially when considering the amount of actual cyberbullying entries, as adding even a second layer to the network already caused

a loss of the most valuable features and ended up degrading the performance. With such small amount of data, it does not seem useful to train deep learning models to solve the classification problem. Still, the dependency based features are showing some potential with CNNs. With a considerably larger dataset it could be possible to outperform other classifiers and especially the traditional approach to word embeddings, namely, using plain tokens, and supplementing them with dependency based features when using deep learning, as was previously proposed by [68].

The experiments show that changing Feature Density in moderate amount can yield good results when using other classifiers than CNNs. However, excessive changes to either too low or too high always showed diminishing results. The threshold was in all cases approximately between 50% and 200% of the original density (TOK), most optimal FDs only slightly varying with each classifier. The exception being Random Forest, which showed a clear spike at around .12 FD. As the usage of high Feature Density datasets showed potential with CNNs, their usage needs to be confirmed with a larger dataset. Also, more exact ideal feature densities need to be confirmed for each classifier using datasets of different sizes and fields to make as accurate ranking of classifiers by FD as possible.

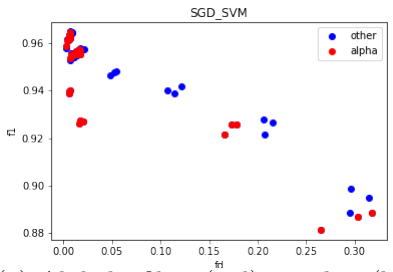
3.3.2.2 Japanese Cyberbullying Dataset

Similarly to the English dataset, I trained all of the classifiers and analyzed the correlation of Feature Density with each of the classifiers using the proposed preprocessing methods. All classification results are represented in Table 3.9 and correlations in Table 3.10 I also excluded preprocessing types that only use POS-tags for the same reason as with the English dataset.

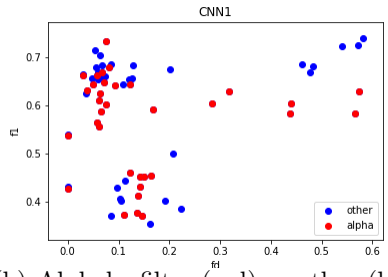
From Table 3.10 one can see that all classifiers have a strong negative correlation with Feature Density, meaning that adding too much linguistic information, such as dependency relations, push the score down. The best results are usually within the range of .08 to .30 FD depending on the classifier. This applies to, for example, the highest performing classifier, one-layer CNN (Figure 3.2), where the maximum classifier performance starts high at around .08 of FD and slightly increases until .30 after which there is a noticeable drop. The performance only falls further as the FD rises, stabilizing after .40 FD. This range includes 43 of the 68 preprocessing methods (Table 3.3). If the weaker feature sets were to be left out, the power

Table 3.7: English Dataset: F1 for all preprocessing types & classifiers; best classifier for each dataset in **bold**; best preprocessing type for each underlined

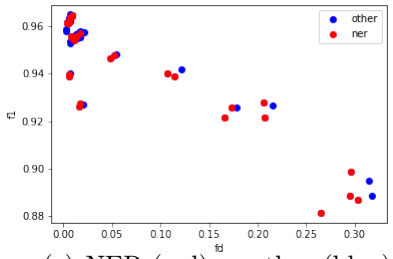
	LBFGS LR	Newton LR	Linear SVM	SGD SVM	KNN	NaiveBayes	RandomForest	AdaBoost	XGBoost	MLP	CNN1	CNN2
CHNK	0.727	0.726	0.718	**0.736**	0.57	0.674	0.613	0.649	0.667	0.724	0.657	0.666
CHNKNERR	0.688	0.695	0.702	0.699	0.58	0.653	0.603	0.608	0.642	**0.704**	0.645	0.662
CHNKNERRALPHA	0.66	0.663	0.651	0.657	0.603	0.626	0.616	0.599	0.653	**0.674**	0.566	0.6
CHNKNERRSTOP	0.686	0.684	0.684	**0.694**	0.577	0.629	0.635	0.621	0.652	0.693	0.402	0.344
CHNKNERRSTOPALPHA	0.618	0.617	0.591	0.607	0.404	0.598	0.62	0.582	**0.648**	0.623	0.451	0.34
CHNKNER	0.718	0.723	0.721	**0.737**	0.582	0.669	0.603	0.63	0.673	0.722	0.654	0.642
CHNKNERALPHA	0.675	0.676	0.663	0.663	0.599	0.641	0.618	0.609	0.649	**0.684**	0.557	0.614
CHNKNERSTOP	**0.724**	**0.724**	0.715	**0.724**	0.582	0.663	0.635	0.652	0.679	0.72	0.501	0.298
CHNKNERSTOPALPHA	0.666	0.661	0.644	**0.668**	0.386	0.615	0.659	0.625	0.656	0.647	0.431	0.406
CHNKALPHA	0.684	0.681	0.669	0.683	0.607	0.643	0.647	0.616	0.676	**0.695**	0.587	0.583
CHNKSTOP	0.722	0.721	0.711	**0.723**	0.577	0.67	0.667	0.648	0.679	0.715	0.386	0.342
CHNKSTOPALPHA	0.629	0.637	0.606	0.619	0.395	0.608	0.649	0.654	**0.664**	0.628	0.455	0.374
DEP	0.617	0.619	0.568	0.587	0.243	0.617	0.536	0.566	0.598	0.594	0.682	**0.694**
DEPNERR	0.61	0.614	0.571	0.587	0.241	0.611	0.533	0.562	0.596	0.595	0.67	**0.695**
DEPNERRALPHA	0.606	0.605	0.589	0.602	0.312	0.596	0.537	0.556	0.595	0.593	0.585	**0.622**
DEPNERRSTOP	0.602	0.599	0.564	0.568	0.273	0.615	0.543	0.572	0.6	0.578	**0.726**	0.702
DEPNERRSTOPALPHA	0.584	0.584	0.56	0.581	0.386	0.599	0.544	0.561	0.595	0.574	0.583	**0.619**
DEPNER	0.624	0.621	0.574	0.585	0.242	0.611	0.528	0.564	0.595	0.592	0.686	**0.692**
DEPNERALPHA	0.585	0.589	0.561	0.579	0.213	0.607	0.578	0.497	0.593	0.603	0.606	**0.623**
DEPNERSTOP	0.611	0.602	0.564	0.576	0.274	0.604	0.527	0.563	0.604	0.577	**0.725**	0.708
DEPNERSTOPALPHA	0.535	0.531	0.523	0.523	0.297	0.543	0.563	0.422	0.576	0.564	0.63	**0.632**
DEPALPHA	0.609	0.612	0.588	0.601	0.314	0.6	0.545	0.552	0.604	0.598	0.606	**0.62**
DEPSTOP	0.606	0.595	0.562	0.571	0.276	0.616	0.544	0.576	0.603	0.584	**0.741**	0.648
DEPSTOPALPHA	0.586	0.587	0.564	0.588	0.388	0.594	0.539	0.568	0.595	0.578	**0.629**	0.625
LEM	0.781	0.786	0.784	**0.79**	0.634	0.715	0.724	0.72	0.744	0.786	0.67	0.665
LEMNERR	0.74	0.737	0.742	0.74	0.601	0.692	0.697	0.683	0.724	**0.749**	0.658	0.663
LEMNERRALPHA	0.729	0.728	0.725	0.725	0.614	0.685	0.699	0.68	0.71	**0.74**	0.645	0.652
LEMNERRSTOP	0.737	0.734	0.726	0.732	0.609	0.682	0.727	0.69	0.72	**0.741**	0.371	0.364
LEMNERRSTOPALPHA	0.732	0.732	0.714	0.727	0.624	0.674	0.723	0.682	0.704	**0.737**	0.372	0.348
LEMPOSS	0.764	0.765	0.769	0.767	0.564	0.713	0.658	0.679	0.717	**0.773**	0.662	0.736
LEMPOSSALPHA	**0.76**	0.758	0.753	0.758	0.406	0.705	0.669	0.674	0.712	0.756	0.603	0.715
LEMPOSSSTOP	0.763	0.766	0.767	**0.774**	0.566	0.709	0.706	0.691	0.72	0.773	0.683	0.725
LEMPOSSSTOPALPHA	0.762	**0.766**	0.748	0.765	0.49	0.702	0.713	0.681	0.714	0.757	0.593	0.716
LEMNER	0.784	0.782	0.787	**0.792**	0.631	0.71	0.716	0.72	0.742	0.78	0.68	0.613
LEMNERALPHA	0.763	0.764	0.765	0.767	0.637	0.699	0.71	0.707	0.742	**0.768**	0.662	0.671
LEMNERSTOP	0.782	0.783	0.782	**0.792**	0.634	0.706	0.745	0.725	0.742	0.78	0.429	0.378
LEMNERSTOPALPHA	**0.77**	0.767	0.752	0.767	0.64	0.693	0.739	0.716	0.738	0.768	0.46	0.414
LEMPOS	0.778	0.778	0.788	**0.79**	0.517	0.711	0.663	0.727	0.741	0.783	0.665	0.64
LEMPOSALPHA	0.768	0.772	0.772	0.768	0.522	0.7	0.654	0.713	0.727	**0.775**	0.664	0.695
LEMPOSSTOP	0.78	0.781	**0.788**	**0.788**	0.642	0.708	0.708	0.721	0.735	0.783	0.715	0.707
LEMPOSSTOPALPHA	0.77	0.769	0.766	0.768	0.669	0.696	0.718	0.722	0.73	**0.778**	0.669	0.698
LEMALPHA	0.755	0.764	0.745	**0.765**	0.294	0.703	0.718	0.705	0.748	0.754	0.61	0.651
LEMSTOP	0.787	0.786	0.784	**0.791**	0.641	0.713	0.754	0.732	0.752	0.789	0.403	0.327
LEMSTOPALPHA	0.772	0.766	0.766	**0.773**	0.357	0.702	0.747	0.712	0.745	0.764	0.377	0.329
POSS	0.487	0.487	0.488	0.491	0.522	0.498	**0.556**	0.509	0.555	0.488	0.54	0.536
POSSALPHA	0.488	0.486	0.488	0.498	0.526	0.498	**0.552**	0.518	0.549	0.493	0.538	0.534
POSSSTOP	0.477	0.477	0.471	0.467	0.518	0.486	**0.54**	0.496	0.533	0.484	0.431	0.434
POSSSTOPALPHA	0.469	0.47	0.471	0.465	0.517	0.478	**0.525**	0.484	0.511	0.491	0.428	0.484
TOK	0.793	0.788	0.793	**0.796**	0.632	0.716	0.711	0.728	0.748	**0.796**	0.659	0.661
TOKNERR	0.741	0.744	0.737	0.743	0.6	0.696	0.688	0.671	0.719	**0.749**	0.655	0.631
TOKNERRALPHA	0.734	0.735	0.735	0.73	0.624	0.683	0.681	0.674	0.704	**0.748**	0.626	0.655
TOKNERRSTOP	0.736	0.736	0.728	0.732	0.609	0.68	0.73	0.678	0.71	**0.751**	0.406	0.317
TOKNERRSTOPALPHA	0.728	0.731	0.727	0.723	0.623	0.675	0.721	0.68	0.698	**0.744**	0.412	0.394
TOKPOSS	0.766	0.768	0.767	**0.783**	0.549	0.715	0.648	0.671	0.715	0.773	0.686	0.729
TOKPOSSALPHA	0.765	0.761	0.763	0.767	0.378	0.709	0.662	0.656	0.709	**0.769**	0.643	0.658
TOKPOSSSTOP	0.763	0.765	0.767	**0.773**	0.563	0.704	0.703	0.684	0.724	0.771	0.675	0.722
TOKPOSSSTOPALPHA	0.774	0.773	0.774	0.771	0.671	0.694	0.722	0.713	0.73	**0.779**	0.68	0.698
TOKNER	**0.789**	0.785	0.788	**0.789**	0.609	0.708	0.703	0.722	0.745	0.784	0.684	0.68
TOKNERALPHA	0.768	0.771	0.763	**0.776**	0.628	0.696	0.701	0.705	0.746	0.775	0.649	0.648
TOKNERSTOP	0.785	**0.791**	0.79	0.79	0.635	0.703	0.732	0.721	0.743	0.79	0.444	0.367
TOKNERSTOPALPHA	0.773	0.771	0.762	**0.774**	0.646	0.691	0.737	0.704	0.74	0.771	0.371	0.379
TOKPOS	0.781	0.783	0.791	**0.798**	0.565	0.713	0.656	0.72	0.739	0.787	0.626	0.705
TOKPOSALPHA	0.775	0.775	0.778	**0.784**	0.576	0.699	0.653	0.705	0.731	0.783	0.633	0.698
TOKPOSSTOP	0.786	0.783	**0.794**	0.792	0.645	0.7	0.711	0.733	0.739	0.789	0.706	0.691
TOKPOSSTOPALPHA	0.759	**0.766**	0.756	0.762	0.458	0.696	0.706	0.679	0.674	0.601	0.734	0.718
TOKALPHA	0.768	0.768	0.757	**0.773**	0.271	0.705	0.721	0.705	0.742	0.756	0.643	0.652
TOKSTOP	0.793	0.79	0.784	**0.794**	0.644	0.708	0.758	0.736	0.749	0.787	0.355	0.321
TOKSTOPALPHA	0.775	**0.776**	0.766	**0.776**	0.342	0.7	0.745	0.714	0.744	0.765	0.452	0.425



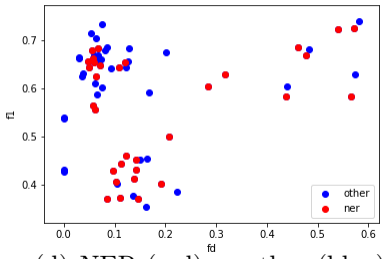
(a) Alphab. filter (red) vs others(blue)



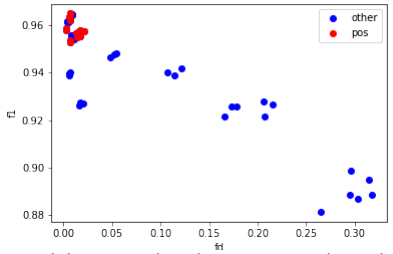
(b) Alphab. filter (red) vs others(blue)



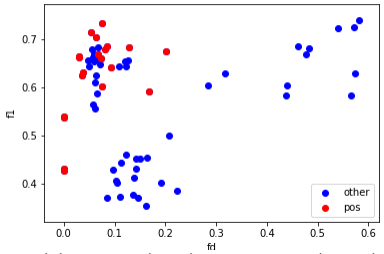
(c) NER (red) vs others(blue)



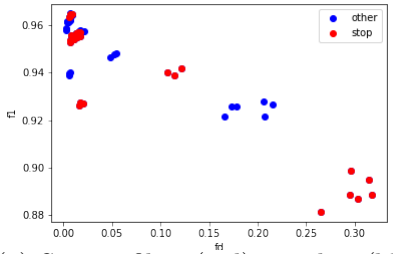
(d) NER (red) vs others(blue)



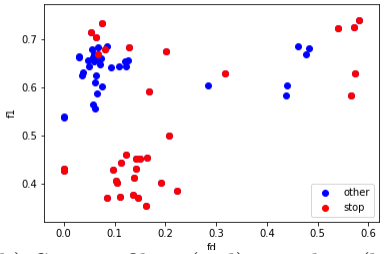
(e) POS (red) vs others(blue)



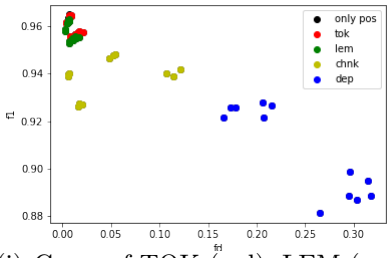
(f) POS (red) vs others(blue)



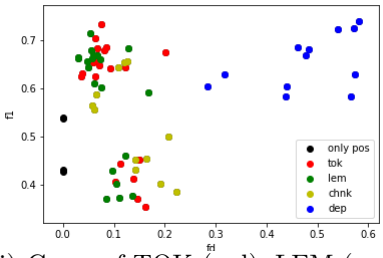
(g) Stopw. filter (red) vs others(blue)



(h) Stopw. filter (red) vs others(blue)



(i) Comp.of TOK (red), LEM (green), CHNK (yellow) and DEP (blue)



(j) Comp.of TOK (red), LEM (green), CHNK (yellow) and DEP (blue)

Figure 3.1: English data: FD & F1 score for SGD SVM (left) and CNN1 (right)

Table 3.8: English Dataset: Classifiers with best F1, preprocessing type and Pearson's correlation coefficient for FD and F1.

Classifier	Best F1	Best PP type	ρ(F1, FD)
SGD SVM	0.798	TOKPOS	-0.8239
MLP	0.7958	TOK	-0.8599
Linear SVM	0.7941	TOKPOSSTOP	-0.834
L-BFGS LR	0.7932	TOKSTOP	-0.8024
Newton LR	0.7915	TOKNERSTOP	-0.8097
RandomForest	0.7582	TOKSTOP	-0.7873
XGBoost	0.7523	LEMSTOP	-0.8303
CNN1	0.7406	DEPSTOP	0.1633
CNN2	0.7357	LEMPOSS	0.0951
AdaBoost	0.7356	TOKSTOP	-0.7362
NaiveBayes	0.7165	TOK	-0.7531
KNN	0.6711	TOKPOSSSTOPALPHA	-0.7116

savings are approximately 5.4kWh for the one-layer CNN, when calculated from Table 3.6.

These results differ from the previous research [16], where CNN had a positive correlation with FD and the highest FD preprocessing methods, namely dependency relations, got the best results. The reason most likely lies in the parser differences (SpaCy used in this research vs. MeCAB[2] used in previous research [16]) and some differences in the network architecture, namely, previous research applied a simple BoW as a language model in CNNs, while I applied a more advanced word embeddings-based language model. The difference is very noticeable as the previous research achieved a score of .927 on the same dataset using CNNs with dependency relations, whereas I only managed to reach 0.802. This suggests two insights. Firstly, it is important to choose the the right and matured intermediary tools for handling the data (MeCab is a well established morphological parser, while the support for Japanese language in SpaCy was added only recently). Secondly, more advanced language models, even if achieving high results in many tasks, might be an overshoot for handling small-sized datasets, like the one used here.

The experiments show that changing Feature Density in moderate amounts could be useful for slightly increasing the F-score. However, excessive changes always showed diminishing results. The threshold was again in all cases approximately between 50% and 200% of the original density (TOK), most optimal FDs varying with each classifier in this range. Also, as the higher FD preprocessings performed

[2] `https://taku910.github.io/mecab/`

quite poorly compared to previous research, I was unable to confirm the positive correlation between classifier performance and FD for the Japanese dataset.

3.3.2.3 Polish Cyberbullying Dataset

Like with the previous two datasets, I analyzed the correlation of Feature Density with each of the classifiers using the proposed preprocessing methods. The classification results are represented in Table 3.12 and correlations in 3.13. Again, I excluded the preprocessing methods that only used POS tags due to poor performance caused by information loss.

From Table 3.13 one can see that most classifiers have a strong negative correlation with Feature Density, meaning that adding too much linguistic information, such as dependency relations, push the score down. The best results are within the range of .08 to .37 of FD depending on the classifier. This range includes 47 of the 68 feature sets (Table 3.4). If the weaker feature sets were to be left out, the power savings are approximately 2.1kWh for training the most expensive classifier, one-layer CNN, calculated from Table 3.6. This result is lower compared to English as cross-validation was not used.

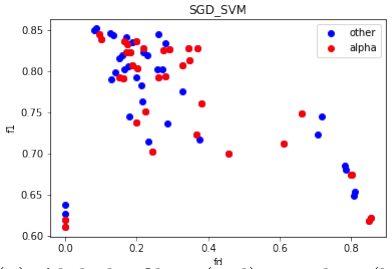
One of the highest performing classifiers, SVM with SGD optimizer (Figure 3.3), indicates that the classifier performance decreases slightly as FD increases within this range. On the other hand, one-layer CNN shows that the maximum classifier performance first increases, peaks at around .20 and then decreases, which shows slight variance between classifiers.

Surprisingly, the performance of Neural Network -based classifiers was poor compared to other datasets. Their F1-score only reached the baseline classifier KNN. The reason behind this could be that Polish is grammatically a lot more complex language compared to, for example, English. The base Feature Density (TOK) of Polish is over double compared to English, which demonstrates this complexity. The linguistic complexity would most likely require more data to make the use of neural networks viable. The effect of FD with neural networks in Polish needs to be explored further using a larger dataset in the future.

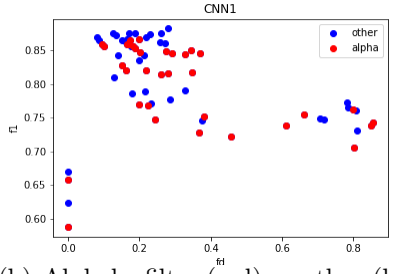
The realization of cyberbullying in the Polish dataset had some differences to English as can be seen from Table 3.11. Polish tended to contain more indirect bullying as the amount of innuendos was almost three times higher than in the

Table 3.9: Japanese Cyberbullying Dataset: F1 for all preprocessing types & classifiers; best classifier for each dataset in **bold**; best preprocessing type for each underlined

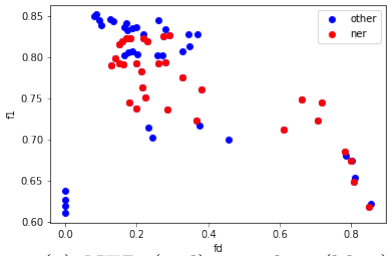
	LBFGS LR	Newton LR	Linear SVM	SGD SVM	KNN	NaiveBayes	RandomForest	AdaBoost	XGBoost	MLP	CNN1	CNN2
CHNK	0.710	0.708	0.718	0.715	0.495	0.753	0.677	0.592	0.632	0.761	**0.771**	0.635
CHNKNERR	0.752	0.751	0.756	0.745	0.517	0.757	0.727	0.690	0.713	0.780	**0.786**	0.699
CHNKNERRALPHA	0.733	0.736	0.744	0.737	0.529	0.745	0.712	0.679	0.703	0.747	**0.770**	0.669
CHNKNERRSTOP	0.745	0.744	0.753	0.737	0.524	0.739	0.729	0.694	0.700	0.775	**0.778**	0.665
CHNKNERRSTOPALPHA	0.731	0.730	**0.741**	0.723	0.539	0.730	0.718	0.676	0.690	0.736	0.727	0.657
CHNKNER	0.758	0.758	0.769	0.763	0.523	0.772	0.738	0.696	0.714	**0.796**	0.789	0.735
CHNKNERALPHA	0.751	0.755	0.757	0.751	0.555	0.767	0.725	0.686	0.712	0.758	**0.768**	0.717
CHNKNERSTOP	0.754	0.755	0.764	0.776	0.543	0.767	0.738	0.696	0.709	0.784	**0.791**	0.768
CHNKNERSTOPALPHA	0.758	0.755	0.760	**0.761**	0.528	0.750	0.721	0.692	0.707	0.745	0.752	0.684
CHNKALPHA	0.714	0.719	0.709	0.702	0.510	0.740	0.684	0.590	0.622	0.738	**0.747**	0.629
CHNKSTOP	0.712	0.714	0.724	0.716	0.506	0.728	0.675	0.580	0.629	**0.753**	0.746	0.688
CHNKSTOPALPHA	0.695	0.701	0.714	0.700	0.518	0.721	0.666	0.572	0.628	0.702	**0.722**	0.604
DEP	0.678	0.681	0.685	0.680	0.495	0.682	0.608	0.534	0.583	0.764	**0.765**	0.573
DEPNERR	0.677	0.684	0.681	0.686	0.479	0.676	0.608	0.520	0.578	**0.791**	0.773	0.590
DEPNERRALPHA	0.667	0.671	0.675	0.674	0.461	0.670	0.585	0.524	0.576	**0.767**	0.762	0.587
DEPNERRSTOP	0.659	0.652	0.659	0.649	0.509	0.645	0.571	0.514	0.549	**0.783**	0.761	0.479
DEPNERRSTOPALPHA	0.617	0.617	0.619	0.618	0.493	0.627	0.534	0.504	0.523	**0.755**	0.738	0.592
DEPNER	0.727	0.727	0.742	0.746	0.502	0.724	0.708	0.690	0.707	**0.792**	0.748	0.547
DEPNERALPHA	0.730	0.731	0.741	0.748	0.427	0.718	0.707	0.689	0.703	**0.782**	0.756	0.575
DEPNERSTOP	0.717	0.716	0.728	0.723	0.544	0.701	0.693	0.681	0.690	**0.802**	0.750	0.654
DEPNERSTOPALPHA	0.715	0.711	0.722	0.713	0.448	0.731	0.695	0.683	0.685	**0.779**	0.739	0.671
DEPALPHA	0.669	0.668	0.681	0.674	0.464	0.674	0.593	0.529	0.571	**0.756**	0.705	0.621
DEPSTOP	0.655	0.658	0.658	0.653	0.507	0.647	0.575	0.523	0.556	**0.776**	0.731	0.659
DEPSTOPALPHA	0.627	0.623	0.622	0.622	0.499	0.628	0.537	0.516	0.535	**0.746**	0.743	0.608
LEM	0.803	0.799	0.817	0.806	0.592	0.823	0.771	0.669	0.723	0.863	**0.868**	0.721
LEMNERR	0.794	0.790	0.796	0.791	0.585	0.808	0.778	0.744	0.763	**0.843**	0.811	0.654
LEMNERRALPHA	0.797	0.796	0.805	0.793	0.585	0.811	0.781	0.748	0.768	**0.845**	0.828	0.708
LEMNERRSTOP	0.787	0.784	0.793	0.793	0.590	0.794	0.796	0.740	0.756	**0.839**	0.836	0.807
LEMNERRSTOPALPHA	0.790	0.791	0.798	0.793	0.587	0.796	0.788	0.743	0.764	0.810	**0.814**	0.776
LEMPOSS	0.815	0.817	0.851	0.842	0.711	0.855	0.785	0.718	0.780	0.866	**0.877**	0.768
LEMPOSSALPHA	0.825	0.823	0.848	0.837	0.734	0.847	0.770	0.690	0.759	0.856	**0.867**	0.572
LEMPOSSSTOP	0.816	0.818	0.850	0.846	0.529	0.851	0.774	0.718	0.764	0.868	**0.876**	0.820
LEMPOSSSTOPALPHA	0.819	0.817	0.841	0.829	0.722	0.844	0.779	0.686	0.743	0.850	**0.851**	0.688
LEMNER	0.806	0.801	0.818	0.816	0.633	0.823	0.790	0.747	0.775	**0.870**	0.865	0.616
LEMNERALPHA	0.809	0.807	0.827	0.823	0.612	0.827	0.797	0.744	0.776	0.857	**0.860**	0.772
LEMNERSTOP	0.799	0.798	0.819	0.823	0.632	0.831	0.802	0.742	0.767	**0.875**	0.870	0.807
LEMNERSTOPALPHA	0.805	0.808	0.830	0.826	0.643	0.834	0.802	0.744	0.769	0.848	**0.849**	0.741
LEMPOS	0.806	0.808	0.845	0.850	0.716	0.830	0.786	0.742	0.768	0.868	**0.870**	0.619
LEMPOSALPHA	0.807	0.803	0.838	0.846	0.704	0.828	0.778	0.728	0.772	0.856	**0.860**	0.663
LEMPOSSTOP	0.802	0.795	0.841	0.847	0.567	0.836	0.770	0.730	0.769	0.868	**0.877**	0.808
LEMPOSSTOPALPHA	0.791	0.789	0.830	0.836	0.700	0.837	0.768	0.707	0.754	0.835	**0.860**	0.783
LEMALPHA	0.800	0.808	0.824	0.807	0.587	0.815	0.758	0.666	0.727	**0.854**	0.853	0.805
LEMSTOP	0.799	0.801	0.813	0.802	0.614	0.812	0.785	0.659	0.723	0.858	**0.862**	0.809
LEMSTOPALPHA	0.803	0.804	0.813	0.807	0.619	0.821	0.776	0.665	0.727	0.842	**0.844**	0.801
POSS	0.645	0.646	0.634	0.627	0.558	0.620	0.645	0.643	0.632	0.645	**0.670**	0.490
POSSALPHA	0.646	0.650	0.635	0.619	0.557	0.618	0.644	0.646	0.640	0.649	**0.658**	0.567
POSSSTOP	0.641	0.643	0.634	0.637	0.553	0.626	0.638	**0.643**	0.630	0.635	0.623	0.492
POSSSTOPALPHA	0.615	**0.616**	0.603	0.610	0.506	0.614	0.610	0.609	0.595	0.614	0.587	0.508
TOK	0.801	0.797	0.813	0.802	0.592	0.817	0.766	0.647	0.724	**0.864**	0.857	0.792
TOKNERR	0.789	0.785	0.797	0.799	0.589	0.798	0.783	0.729	0.756	0.838	**0.844**	0.793
TOKNERRALPHA	0.791	0.792	0.802	0.792	0.567	0.797	0.778	0.737	0.759	**0.836**	0.821	0.722
TOKNERRSTOP	0.776	0.778	0.795	0.783	0.582	0.788	0.776	0.727	0.756	0.835	**0.843**	0.709
TOKNERRSTOPALPHA	0.785	0.787	0.796	0.795	0.571	0.783	0.778	0.736	0.761	**0.818**	0.817	0.742
TOKPOSS	0.815	0.812	0.843	0.835	0.695	0.849	0.775	0.719	0.756	0.876	**0.876**	0.683
TOKPOSSALPHA	0.788	0.789	0.829	0.833	0.716	0.823	0.752	0.715	0.748	0.864	**0.865**	0.646
TOKPOSSSTOP	0.823	0.814	0.834	0.828	0.713	0.848	0.767	0.701	0.760	**0.855**	0.821	0.643
TOKPOSSSTOPALPHA	0.817	0.822	0.837	0.835	0.518	0.846	0.764	0.703	0.764	0.876	0.883	0.759
TOKNER	0.814	0.812	0.830	0.828	0.718	0.835	0.763	0.667	0.731	0.846	**0.847**	0.686
TOKNERALPHA	0.798	0.800	0.816	0.820	0.610	0.825	0.790	0.734	0.762	**0.870**	0.864	0.834
TOKNERSTOP	0.803	0.805	0.825	0.824	0.608	0.822	0.790	0.745	0.768	0.842	**0.857**	0.811
TOKNERSTOPALPHA	0.794	0.794	0.819	0.820	0.616	0.820	0.797	0.735	0.759	0.878	0.875	0.849
TOKPOS	0.799	0.796	0.822	0.828	0.630	0.826	0.794	0.730	0.763	0.846	**0.847**	0.773
TOKPOSALPHA	0.808	0.807	0.844	0.852	0.706	0.828	0.777	0.724	0.770	**0.870**	0.865	0.674
TOKPOSSTOP	0.802	0.800	0.839	0.840	0.691	0.823	0.771	0.715	0.763	0.839	**0.857**	0.615
TOKPOSSTOPALPHA	0.795	0.794	0.834	0.844	0.560	0.832	0.766	0.720	0.762	0.863	**0.873**	0.661
TOKALPHA	0.801	0.799	0.807	0.804	0.584	0.817	0.761	0.655	0.712	0.847	**0.848**	0.798
TOKSTOP	0.803	0.801	0.815	0.803	0.584	0.809	0.778	0.646	0.717	0.857	**0.861**	0.782
TOKSTOPALPHA	0.800	0.800	0.811	0.813	0.591	0.821	0.779	0.650	0.717	**0.827**	0.818	0.710



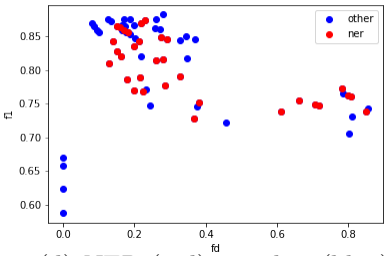
(a) Alphab. filter (red) vs others(blue)



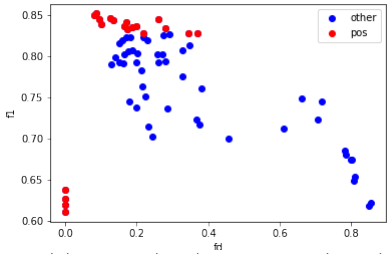
(b) Alphab. filter (red) vs others(blue)



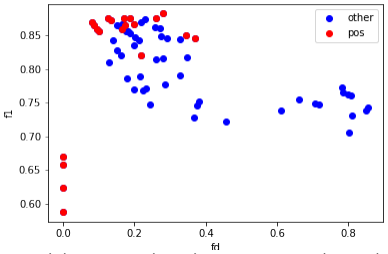
(c) NER (red) vs others(blue)



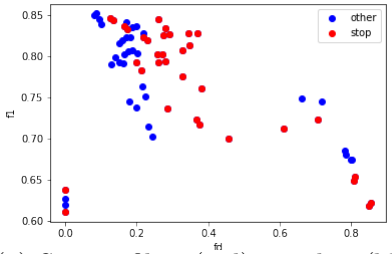
(d) NER (red) vs others(blue)



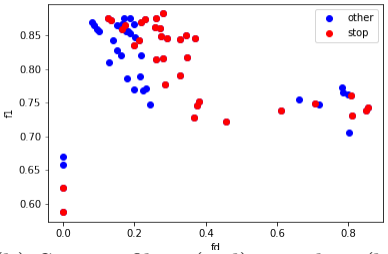
(e) POS (red) vs others(blue)



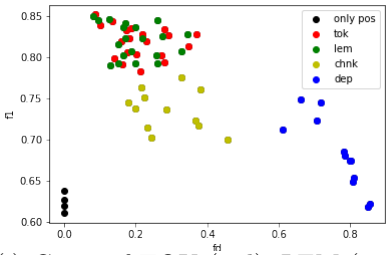
(f) POS (red) vs others(blue)



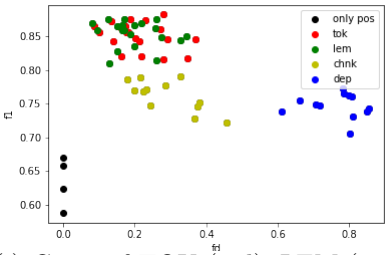
(g) Stopw. filter (red) vs others(blue)



(h) Stopw. filter (red) vs others(blue)



(i) Comp.of TOK (red), LEM (green), CHNK (yellow) and DEP (blue)



(j) Comp.of TOK (red), LEM (green), CHNK (yellow) and DEP (blue)

Figure 3.2: Japanese Data: FD & F1 score for SGD SVM (left) and CNN1 (right)

Table 3.10: Japanese Dataset: Classifiers with best F1, preprocessing type and Pearson's correlation coefficient for FD and F1.

Classifier	Best F1	Best PP type	ρ(F1, FD)
CNN1	0.8829	TOKPOSSTOP	-0.7356
MLP	0.8776	TOKNERSTOP	-0.6107
NaiveBayes	0.8552	LEMPOS	-0.8675
SGD SVM	0.8523	TOKPOSS	-0.8270
Linear SVM	0.8508	LEMPOS	-0.8415
CNN2	0.8488	TOKNERRSTOP	-0.5637
L-BFGSLR	0.8251	LEMPOSALPHA	-0.8506
Newton LR	0.8227	LEMPOSALPHA	-0.8511
RandomForest	0.8022	LEMNERSTOPALPHA	-0.8711
XGBoost	0.7801	LEMPOS	-0.8461
AdaBoost	0.7484	LEMNERRALPHA	-0.7902
KNN	0.7339	LEMPOSALPHA	-0.6320

Table 3.11: Categories of cyberbullying in English and Polish datasets

CB category	**English #**	**Polish #**	**English %**	**Polish %**
Threat	29	25	3.2	2.5
Innuendo	24	68	2.6	6.9
Sexual harassment	92	N/A	10	N/A
Insult	198	459	21.6	46.6
Blackmail	5	0	0.5	0
Mockery	338	471	36.9	42.3
Phishing, revealing private information	0	23	0	2.3
Vulgarism	135	129	14.7	13.1

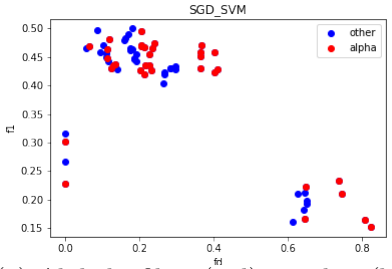
English dataset. The Polish dataset also contained some revealing of personal information while the English dataset had none. Also, Polish contained more personal insults than the English dataset. On the other hand, the English dataset contained some blackmailing, while Polish did not have any. The fact that Polish had more indirect bullying in the form of innuendos compared to English and that Polish language is more complex than English could be the reasons for the overall lower performance of the classifiers on the Polish dataset.

3.3.2.4 Verification Dataset

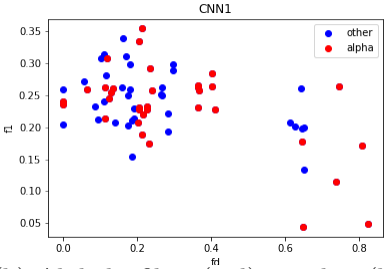
Also for the verification dataset, I analyzed the correlation of Feature Density with each of the classifiers using the proposed preprocessing methods. The classification scores are shown in Table 3.14 and correlations in Table 3.15. After excluding the preprocessing methods that only used POS tags, one can see that Logistic Regression and SVMs have a strong negative correlation with Feature

Table 3.12: Polish Dataset: F1 for all preprocessing types & classifiers; best classifier for each dataset in **bold**; best preprocessing type for each underlined

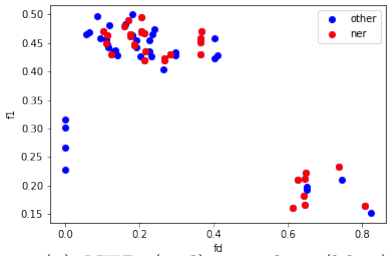
	LBFGS LR	Newton LR	Linear SVM	SGD SVM	KNN	NaiveBayes	RandomForest	AdaBoost	XGBoost	MLP	CNN1	CNN2
CHNK	0.437	0.419	0.424	0.443	0.163	**0.455**	0.124	0.308	0.278	0.245	0.213	0.270
CHNKNERR	0.448	0.433	0.455	**0.462**	0.175	0.446	0.099	0.272	0.183	0.234	0.249	0.292
CHNKNERRALPHA	0.444	0.440	0.446	**0.469**	0.201	0.468	0.085	0.291	0.205	0.224	0.228	0.211
CHNKNERRSTOP	**0.447**	0.414	0.408	0.421	0.231	0.400	0.187	0.333	0.299	0.223	0.250	0.312
CHNKNERRSTOPALPHA	0.425	0.427	0.425	**0.452**	0.241	0.401	0.212	0.283	0.246	0.276	0.262	0.187
CHNKNER	0.439	0.435	0.416	0.445	0.170	**0.459**	0.113	0.304	0.205	0.271	0.155	0.192
CHNKNERALPHA	0.444	0.437	0.424	0.419	0.182	**0.459**	0.086	0.265	0.236	0.290	0.189	0.183
CHNKNERSTOP	0.417	0.414	0.396	**0.430**	0.225	0.413	0.176	0.308	0.295	0.254	0.194	0.307
CHNKNERSTOPALPHA	0.459	**0.462**	0.436	0.431	0.213	0.441	0.174	0.325	0.309	0.265	0.231	0.260
CHNKALPHA	0.435	0.452	0.412	0.435	0.180	**0.457**	0.085	0.256	0.192	0.259	0.228	0.268
CHNKSTOP	0.448	**0.453**	0.394	0.433	0.199	0.403	0.188	0.388	0.331	0.267	0.298	0.307
CHNKSTOPALPHA	0.450	**0.459**	0.441	0.459	0.225	0.438	0.208	0.330	0.266	0.279	0.264	0.309
DEP	0.316	0.316	0.204	0.199	0.235	**0.346**	0.029	0.250	0.070	0.125	0.200	0.123
DEPNERR	0.322	0.312	0.214	0.211	0.243	**0.340**	0.029	0.242	0.057	0.128	0.197	0.057
DEPNERRALPHA	0.261	0.251	0.200	0.233	0.263	**0.335**	0.043	0.260	0.071	0.176	0.115	0.213
DEPNERRSTOP	0.276	0.290	0.195	0.182	0.250	**0.301**	0.029	0.195	0.043	0.122	0.260	0.055
DEPNERRSTOPALPHA	0.159	0.158	0.154	0.165	0.260	**0.264**	0.058	0.119	0.057	0.158	0.171	0.027
DEPNER	0.345	0.339	0.191	0.210	0.224	**0.345**	0.043	0.269	0.068	0.140	0.201	0.044
DEPNERALPHA	0.250	0.240	0.201	0.222	0.193	**0.329**	0.044	0.230	0.043	0.111	0.044	0.000
DEPNERSTOP	**0.303**	0.299	0.173	0.160	0.221	0.289	0.044	0.178	0.042	0.077	0.206	0.146
DEPNERSTOPALPHA	0.172	0.181	0.166	0.167	0.175	**0.239**	0.058	0.093	0.056	0.201	0.177	0.168
DEPALPHA	0.230	0.251	0.200	0.210	0.260	**0.358**	0.043	0.234	0.071	0.197	0.263	0.133
DEPSTOP	0.273	0.289	0.184	0.193	0.240	**0.304**	0.043	0.215	0.070	0.111	0.134	0.014
DEPSTOPALPHA	0.155	0.168	0.176	0.152	0.256	**0.276**	0.071	0.105	0.044	0.106	0.049	0.135
LEM	**0.481**	0.473	0.428	0.444	0.281	0.460	0.125	0.361	0.317	0.285	0.281	0.275
LEMNERR	0.466	0.456	0.462	**0.471**	0.273	0.452	0.113	0.309	0.244	0.277	0.308	0.307
LEMNERRALPHA	0.455	0.449	0.452	0.464	0.239	**0.478**	0.112	0.341	0.214	0.324	0.262	0.233
LEMNERRSTOP	0.484	0.484	0.494	0.479	0.302	0.417	0.221	0.347	0.310	0.306	0.263	0.283
LEMNERRSTOPALPHA	0.482	0.484	0.471	**0.496**	0.272	0.439	0.211	0.364	0.287	0.296	0.333	0.312
LEMPOSS	0.494	0.483	0.444	0.456	0.265	0.459	0.112	0.390	0.214	0.287	0.313	0.190
LEMPOSSALPHA	**0.467**	0.462	0.421	0.438	0.287	0.444	0.125	0.354	0.221	0.279	0.260	0.187
LEMPOSSSTOP	0.467	0.451	0.457	**0.483**	0.322	0.437	0.237	0.448	0.345	0.297	0.338	0.241
LEMPOSSSTOPALPHA	0.490	**0.498**	0.448	0.474	0.352	0.426	0.255	0.421	0.385	0.312	0.258	0.288
LEMNER	**0.466**	0.465	0.419	0.450	0.281	0.464	0.125	0.381	0.241	0.267	0.239	0.237
LEMNERALPHA	**0.487**	0.464	0.414	0.431	0.225	0.482	0.099	0.386	0.232	0.275	0.245	0.265
LEMNERSTOP	0.461	**0.494**	0.467	0.490	0.301	0.430	0.224	0.391	0.378	0.348	0.310	0.334
LEMNERSTOPALPHA	**0.486**	0.473	0.477	0.467	0.235	0.461	0.232	0.391	0.343	0.346	0.354	0.315
LEMPOS	0.447	0.457	**0.472**	0.465	0.271	0.440	0.099	0.427	0.219	0.266	0.271	0.120
LEMPOSALPHA	**0.484**	0.473	0.420	0.469	0.286	0.462	0.110	0.361	0.186	0.275	0.258	0.193
LEMPOSSTOP	0.461	0.457	0.491	**0.498**	0.341	0.422	0.174	0.438	0.259	0.305	0.232	0.164
LEMPOSSTOPALPHA	0.449	**0.498**	0.466	0.481	0.357	0.419	0.192	0.438	0.256	0.325	0.307	0.246
LEMALPHA	0.491	**0.493**	0.440	0.436	0.225	0.462	0.112	0.381	0.220	0.315	0.254	0.250
LEMSTOP	0.490	**0.500**	0.469	0.500	0.281	0.430	0.231	0.408	0.383	0.305	0.299	0.264
LEMSTOPALPHA	0.494	0.502	0.452	0.465	0.242	0.453	0.231	0.425	0.371	0.330	0.292	0.319
POSS	0.313	0.314	0.298	**0.315**	0.184	0.301	0.141	0.293	0.172	0.293	0.204	0.198
POSSALPHA	0.310	**0.316**	0.297	0.301	0.180	0.301	0.113	0.298	0.179	0.282	0.235	0.170
POSSSTOP	0.295	**0.297**	0.288	0.266	0.188	0.251	0.193	0.282	0.223	0.257	0.258	0.230
POSSSTOPALPHA	0.256	0.246	0.247	0.229	0.078	0.239	0.203	0.264	0.240	0.246	0.240	**0.267**
TOK	0.437	0.444	0.422	**0.456**	0.166	0.455	0.099	0.303	0.171	0.278	0.229	0.248
TOKNERR	0.440	0.444	0.432	**0.465**	0.178	0.455	0.072	0.266	0.194	0.245	0.203	0.212
TOKNERRALPHA	0.424	0.432	0.443	**0.472**	0.190	0.451	0.099	0.275	0.181	0.266	0.230	0.262
TOKNERRSTOP	0.425	**0.430**	0.406	0.424	0.244	0.384	0.150	0.314	0.267	0.226	0.262	0.288
TOKNERRSTOPALPHA	0.434	0.432	0.425	**0.458**	0.229	0.408	0.151	0.284	0.267	0.242	0.265	0.236
TOKPOSS	0.443	0.441	0.450	**0.464**	0.264	0.461	0.071	0.321	0.205	0.201	0.258	0.056
TOKPOSSALPHA	0.393	0.400	0.403	**0.427**	0.317	0.411	0.185	0.317	0.251	0.208	0.208	0.040
TOKPOSSSTOP	0.443	0.449	0.424	0.427	0.273	**0.457**	0.086	0.246	0.160	0.253	0.174	0.173
TOKPOSSSTOPALPHA	**0.430**	0.420	0.400	0.404	0.306	0.416	0.137	0.345	0.277	0.190	0.251	0.212
TOKNER	0.429	**0.442**	0.425	0.429	0.327	0.432	0.197	0.297	0.263	0.284	0.227	0.257
TOKNERALPHA	0.444	0.439	0.416	0.448	0.170	**0.453**	0.085	0.314	0.191	0.309	0.210	0.188
TOKNERSTOP	0.447	0.435	0.428	0.435	0.176	**0.461**	0.085	0.282	0.192	0.267	0.220	0.162
TOKNERSTOPALPHA	0.429	0.427	0.398	**0.430**	0.242	0.405	0.174	0.335	0.291	0.271	0.221	0.281
TOKPOS	0.451	**0.473**	0.445	0.471	0.225	0.438	0.199	0.316	0.298	0.272	0.257	0.259
TOKPOSALPHA	0.422	0.434	0.421	**0.459**	0.255	0.450	0.058	0.302	0.171	0.271	0.212	0.071
TOKPOSSTOP	0.434	0.436	0.429	**0.447**	0.266	0.442	0.043	0.259	0.168	0.213	0.214	0.129
TOKPOSSTOPALPHA	0.388	0.393	0.404	**0.428**	0.286	0.398	0.138	0.300	0.264	0.208	0.208	0.040
TOKALPHA	0.442	0.446	0.414	**0.456**	0.176	0.448	0.112	0.303	0.215	0.261	0.232	0.286
TOKSTOP	0.438	**0.440**	0.392	0.429	0.220	0.423	0.188	0.387	0.287	0.234	0.288	0.352
TOKSTOPALPHA	**0.463**	0.448	0.434	0.423	0.217	0.451	0.172	0.335	0.271	0.344	0.283	0.276



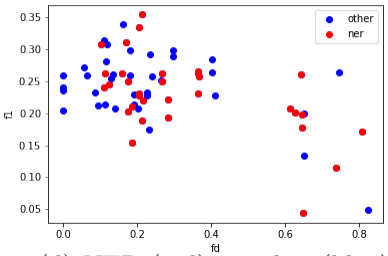
(a) Alphab. filter (red) vs others(blue)



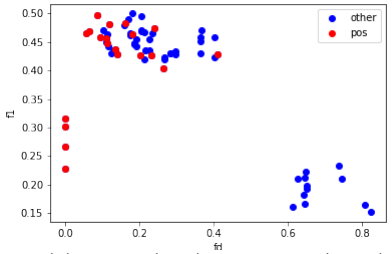
(b) Alphab. filter (red) vs others(blue)



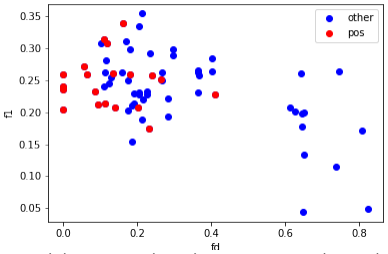
(c) NER (red) vs others(blue)



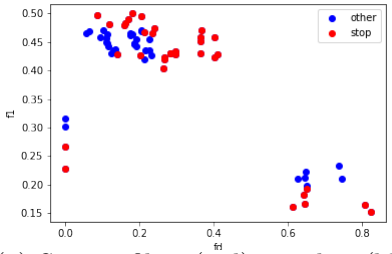
(d) NER (red) vs others(blue)



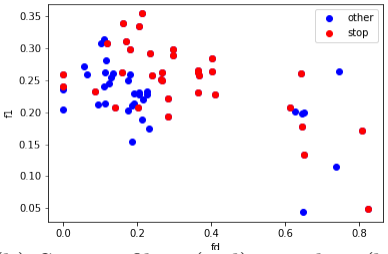
(e) POS (red) vs others(blue)



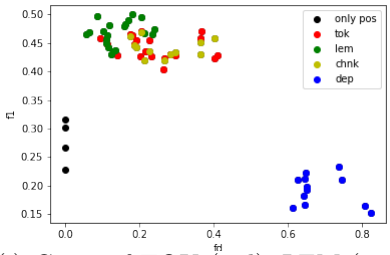
(f) POS (red) vs others(blue)



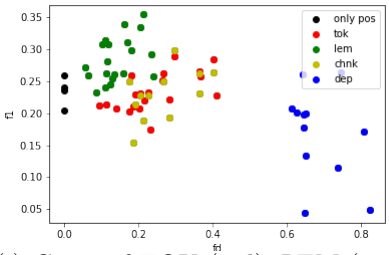
(g) Stopw. filter (red) vs others(blue)



(h) Stopw. filter (red) vs others(blue)



(i) Comp.of TOK (red), LEM (green), CHNK (yellow) and DEP (blue)



(j) Comp.of TOK (red), LEM (green), CHNK (yellow) and DEP (blue)

Figure 3.3: Polish Data: FD & F1 score for SGD SVM (left) and CNN1 (right)

Table 3.13: Polish Dataset: Classifiers with best F1, preprocessing type and Pearson's correlation coefficient for FD and F1.

Classifier	Best F1	Best PP type	ρ(F1, FD)
Newton LR	0.5021	LEMstopalpha	-0.8824
SGD SVM	0.5000	LEMstop	-0.9154
L-BFGS LR	0.4944	LEMPOS	-0.8869
Linear SVM	0.4936	LEMNERRSTOP	-0.9117
NaiveBayes	0.4819	LEMNERALPHA	-0.8719
AdaBoost	0.4480	LEMPOSSTOP	-0.7298
XGBoost	0.3846	LEMPOSSTOPALPHA	-0.6749
KNN	0.3571	LEMPOSSSTOPALPHA	-0.1470
CNN1	0.3535	LEMNERSTOPALPHA	-0.5461
CNN2	0.3517	TOKSTOP	-0.4314
MLP	0.3483	LEMNERSTOP	-0.7447
RandomForest	0.2548	LEMPOSSTOPALPHA	-0.4224

Density. So these classifiers seem to have a weaker performance if a lot of linguistic information is added, and the best results being usually within the range of .002 to .015 FD depending on the classifier. This range includes 37 of the 68 preprocessing methods (Table 3.5). This can be seen from, for example, the highest performing classifier, SVM with SGD optimizer (Figure 3.4), where the maximum classifier performance increases until peaking at .007 after which there is a noticeable drop. The performance only falls further as the FD rises. If the feature sets outside this range would be left out, the power savings are approximately 400Wh for the SGD SVM, when calculated from Table 3.6.

For CNNs however, there was no correlation between FD and the classifier performance, with the higher FD datasets performing almost equally when comparing to the low FD dataset, similarly as it was for English Cyberbullying Dataset, suggesting this could be a more general characteristics of the language itself. Taking a look at one layer CNN's performance, which was better than the CNN with two layers, Figure 3.4 shows that the performance stays quite stable throughout the whole range of feature densities. The scores are also generally much higher due to the classification problem (sentiment analysis of 1- and 5-start reviews) being much simpler compared to cyberbullying. All of the classifiers got considerably high results across feature densities without extreme changes, unlike other datasets. In other words, all of the classifiers seem to have reached their maximum performance for the data and there is no more score to gain without overfitting. Due to this, I am unable to estimate the possible time save by feature density in this case,

meaning that for simple tasks such as this one it does not matter that much which classifier or which preprocessing is used, as it will still reach a considerably high performance.

3.3.3 Analysis of Linguistically-backed Preprocessing

3.3.3.1 English Cyberbullying Dataset

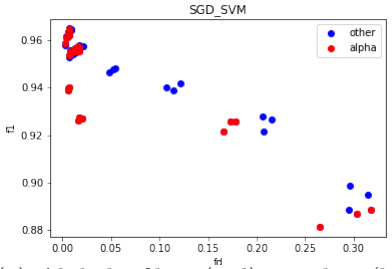
From the results it can be seen that most of the classifiers scored highest or close to highest on pure tokens. CNNs also performed quite well on tokens, but also on dependency-based preprocessings. Using lemmas generally got slightly lower scores than tokens presumably due to information loss. Chunking got low performance overall and was clearly outperformed by dependency-based features in CNNs. Using only parts-of-speech tags achieved very low performance and thus it should be only used as a supplement to other preprocessing methods. However, the simplest preprocessing, namely, tokens were usually among the highest scoring preprocessing type, this provides the first known empirical proof for using simple words as a default resource for embeddings in practically all previous research on word embeddings. However, it was still possible to outperform simple tokens by a number of preprocessing combinations.

Stopword filtering seemed to be the one of the most effective preprocessing techniques for traditional classifiers, which can be seen from Table 3.8 as it was used in the majority of the highest scores. The problem with stopword filtering was that the scores fluctuated a lot, having both low and very high scores and scoring high mostly with Logistic Regression and all of the tree based classifiers. An important thing to note is that the preprocessing method had extremely polarized performance with CNNs, scoring either very high or low. Overall, stopword filtering yielded the most top scores of any preprocessing method considering all the classifiers.

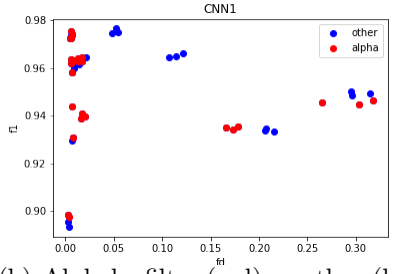
Another very effective preprocessing method was Parts of Speech merging (POS), which achieved high performance overall when added to TOK or LEM. The method also got the highest scores with multiple classifiers, especially SVMs. Adding parts-of-speech information to the respective words usually achieved a higher score than using them as a separate feature. This keeps the information directly connected to the word itself, which seems a better option from the point of view of information

Table 3.14: Verification Dataset: F1 for all preprocessing types & classifiers; best classifier for each dataset in **bold**; best preprocessing type for each underlined

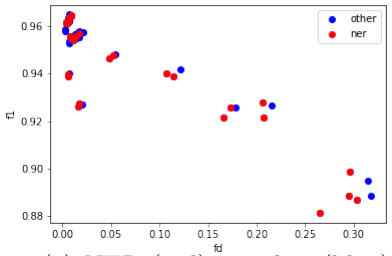
	LBFGS LR	SGD SVM	KNN	NaiveBayes	RandomForest	XGBoost	MLP	CNN1	CNN2
CHNK	0.958	0.948	0.791	0.924	0.909	0.918	0.968	**0.975**	0.971
CHNKNERR	0.956	0.947	0.785	0.925	0.908	0.922	0.967	**0.975**	0.972
CHNKNERRALPHA	0.945	0.939	0.783	0.928	0.899	0.916	0.950	**0.962**	0.958
CHNKNERRSTOP	0.949	0.940	0.759	0.935	0.923	0.901	**0.967**	0.965	0.965
CHNKNERRSTOPALPHA	0.933	0.926	0.609	0.921	0.905	0.891	**0.943**	0.939	0.942
CHNKNER	0.957	0.948	0.783	0.923	0.909	0.922	0.967	0.977	0.972
CHNKNERALPHA	0.947	0.940	0.781	0.929	0.904	0.918	0.953	**0.964**	0.960
CHNKNERSTOP	0.950	0.939	0.767	0.933	0.923	0.902	**0.968**	0.965	0.965
CHNKNERSTOPALPHA	0.935	0.927	0.611	0.923	0.909	0.893	**0.945**	0.941	0.943
CHNKALPHA	0.947	0.940	0.786	0.930	0.902	0.916	0.953	**0.962**	0.959
CHNKSTOP	0.951	0.942	0.769	0.934	0.925	0.898	0.969	0.966	0.964
CHNKSTOPALPHA	0.935	0.927	0.609	0.924	0.908	0.893	0.944	0.939	**0.944**
DEP	**0.948**	0.927	0.777	0.940	0.900	0.878	0.920	0.933	0.933
DEPNERR	**0.949**	0.928	0.777	0.941	0.901	0.881	0.921	0.934	0.934
DEPNERRALPHA	**0.945**	0.926	0.773	0.941	0.899	0.869	0.924	0.934	0.934
DEPNERRSTOP	0.928	0.898	0.726	0.946	0.891	0.846	0.946	**0.948**	0.947
DEPNERRSTOPALPHA	0.916	0.887	0.732	0.935	0.885	0.824	0.943	**0.945**	0.942
DEPNER	**0.946**	0.922	0.774	0.940	0.898	0.881	0.923	0.935	0.933
DEPNERALPHA	**0.943**	0.921	0.032	0.940	0.897	0.876	0.926	0.935	0.934
DEPNERSTOP	0.924	0.889	0.729	0.945	0.893	0.847	0.944	**0.950**	0.945
DEPNERSTOPALPHA	0.914	0.881	0.155	0.934	0.884	0.831	0.944	**0.945**	0.944
DEPALPHA	**0.944**	0.926	0.773	0.940	0.898	0.869	0.926	0.935	0.935
DEPSTOP	0.927	0.895	0.726	0.945	0.889	0.842	0.947	**0.949**	0.944
DEPSTOPALPHA	0.915	0.888	0.735	0.934	0.885	0.815	0.945	**0.946**	0.942
LEM	0.968	0.963	0.768	0.924	0.934	0.948	0.966	**0.974**	0.970
LEMNERR	0.967	0.962	0.772	0.925	0.934	0.948	0.963	**0.972**	0.970
LEMNERRALPHA	0.966	0.961	0.774	0.924	0.934	0.947	0.964	**0.973**	0.969
LEMNERRSTOP	0.960	0.954	0.766	0.920	0.939	0.936	**0.965**	0.962	0.963
LEMNERRSTOPALPHA	0.961	0.955	0.766	0.920	0.939	0.936	**0.964**	0.963	0.963
LEMPOSS	**0.968**	0.962	0.744	0.925	0.932	0.947	0.944	0.958	0.958
LEMPOSSALPHA	**0.967**	0.962	0.764	0.925	0.932	0.946	0.948	0.958	0.956
LEMPOSSSTOP	**0.962**	0.955	0.740	0.921	0.938	0.933	0.962	0.962	0.958
LEMPOSSSTOPALPHA	0.962	0.955	0.760	0.920	0.940	0.932	0.962	**0.963**	0.960
LEMNER	0.968	0.963	0.765	0.922	0.935	0.948	0.963	**0.973**	0.971
LEMNERALPHA	0.967	0.962	0.763	0.922	0.934	0.947	0.964	**0.973**	0.969
LEMNERSTOP	0.961	0.955	0.763	0.918	0.939	0.938	**0.967**	0.962	0.963
LEMNERSTOPALPHA	0.962	0.956	0.761	0.918	0.938	0.937	**0.965**	0.963	0.963
LEMPOS	**0.967**	0.958	0.741	0.923	0.927	0.949	0.806	0.895	0.891
LEMPOSALPHA	**0.966**	0.959	0.737	0.923	0.928	0.948	0.818	0.898	0.894
LEMPOSSTOP	**0.961**	0.953	0.765	0.919	0.935	0.937	0.912	0.930	0.930
LEMPOSSTOPALPHA	**0.961**	0.954	0.769	0.919	0.938	0.936	0.940	0.944	0.948
LEMALPHA	0.967	0.963	0.765	0.923	0.935	0.947	0.966	**0.972**	0.968
LEMSTOP	0.962	0.956	0.763	0.919	0.940	0.937	**0.967**	0.963	0.964
LEMSTOPALPHA	0.962	0.956	0.762	0.918	0.941	0.936	**0.965**	0.963	0.963
POSS	0.744	0.742	0.658	0.690	0.747	0.750	0.668	0.836	**0.841**
POSSALPHA	0.744	0.749	0.658	0.690	0.746	0.750	0.668	0.838	**0.843**
POSSSTOP	0.723	0.729	0.613	0.691	0.715	0.728	0.666	0.777	**0.778**
POSSSTOPALPHA	0.712	0.713	0.504	0.675	0.695	0.718	0.600	0.755	**0.764**
TOK	0.970	0.965	0.791	0.928	0.934	0.949	0.967	**0.974**	0.971
TOKNERR	0.969	0.964	0.787	0.929	0.931	0.948	0.967	**0.975**	0.971
TOKNERRALPHA	0.969	0.964	0.787	0.928	0.935	0.948	0.966	**0.974**	0.970
TOKNERRSTOP	0.962	0.957	0.763	0.924	0.941	0.935	**0.967**	0.964	0.965
TOKNERRSTOPALPHA	0.962	0.957	0.763	0.923	0.940	0.936	**0.968**	0.964	0.965
TOKPOSS	0.970	0.965	0.755	0.930	0.931	0.946	0.947	0.960	0.960
TOKPOSSALPHA	**0.963**	0.955	0.771	0.924	0.939	0.939	0.916	0.931	0.933
TOKPOSSSTOP	**0.970**	0.965	0.783	0.929	0.933	0.946	0.951	0.961	0.957
TOKPOSSSTOPALPHA	0.963	0.957	0.730	0.927	0.941	0.933	0.963	**0.965**	0.959
TOKNER	0.964	0.958	0.760	0.927	0.942	0.933	**0.967**	0.964	0.962
TOKNERALPHA	0.970	0.965	0.783	0.926	0.935	0.950	0.966	**0.974**	0.972
TOKNERSTOP	0.970	0.965	0.781	0.925	0.936	0.949	0.966	**0.974**	0.971
TOKNERSTOPALPHA	0.963	0.957	0.765	0.923	0.941	0.937	**0.968**	0.963	0.965
TOKPOS	0.962	0.957	0.762	0.923	0.942	0.937	**0.968**	0.964	0.966
TOKPOSALPHA	**0.969**	0.962	0.752	0.927	0.929	0.951	0.806	0.893	0.892
TOKPOSSTOP	**0.969**	0.962	0.753	0.926	0.930	0.950	0.825	0.897	0.895
TOKPOSSTOPALPHA	**0.963**	0.956	0.761	0.925	0.938	0.938	0.916	0.931	0.933
TOKALPHA	0.969	0.964	0.787	0.927	0.937	0.949	0.968	**0.974**	0.971
TOKSTOP	0.964	0.958	0.765	0.923	0.943	0.936	**0.968**	0.963	0.966
TOKSTOPALPHA	0.963	0.958	0.764	0.923	0.944	0.937	**0.968**	0.964	0.965



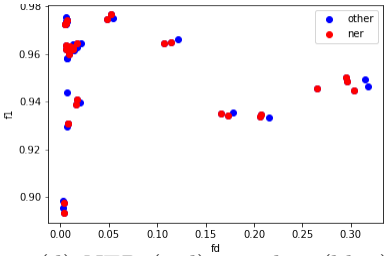
(a) Alphab. filter (red) vs others(blue)



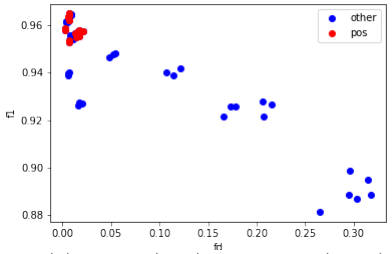
(b) Alphab. filter (red) vs others(blue)



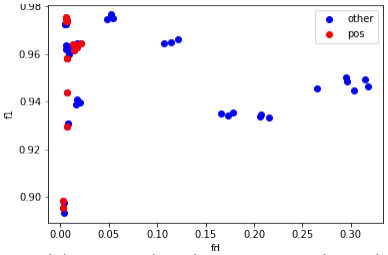
(c) NER (red) vs others(blue)



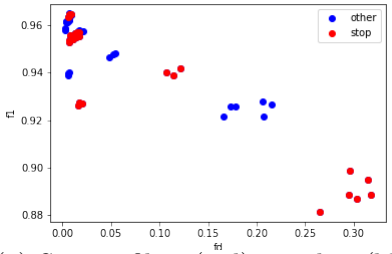
(d) NER (red) vs others(blue)



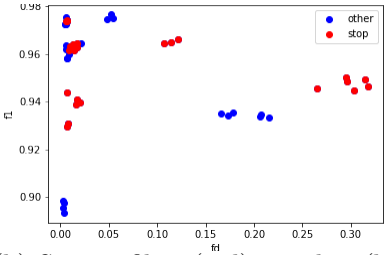
(e) POS (red) vs others(blue)



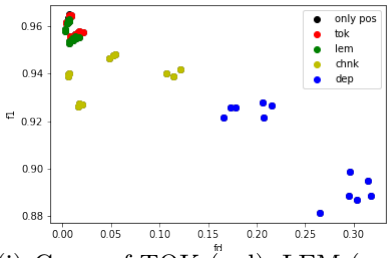
(f) POS (red) vs others(blue)



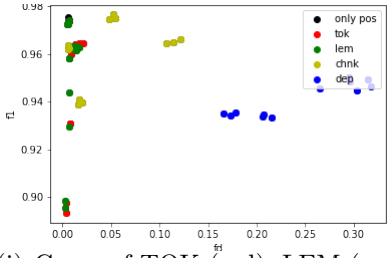
(g) Stopw. filter (red) vs others(blue)



(h) Stopw. filter (red) vs others(blue)



(i) Comp.of TOK (red), LEM (green), CHNK (yellow) and DEP (blue)



(j) Comp.of TOK (red), LEM (green), CHNK (yellow) and DEP (blue)

Figure 3.4: Verif. Data: FD & F1 score for SGD SVM (left) and CNN1 (right)

Table 3.15: Verification Dataset: Classifiers with best F1, preprocessing type and Pearson's correlation coefficient for FD and F1.

Classifier	Best F1	Best PP type	ρ(F1, FD)
CNN1	0.9766	CHNKNER	-0.2131
CNN2	0.9723	CHNKNERR	-0.2365
L-BFGS LR	0.9704	TOKPOS	-0.8243
MLP	0.9689	CHNKSTOP	-0.0410
SGD SVM	0.9650	TOK	-0.9127
XGBoost	0.9506	TOKPOSS	-0.9345
NaiveBayes	0.9456	DEPNERSTOP	0.8452
RandomForest	0.9435	TOKSTOPALPHA	-0.8041
KNN	0.7911	CHNK	-0.3000

preservation.

Using Named Entity Recognition information reduced the classifier performance most of the time, only achieving a high score with one classifier, Newton-LR. The performance of using NER seemed clearly inferior compared to stopword filtering or parts-of-speech information. Replacing words with their NER information seems to cause too much information loss and reduces the performance when comparing to plain tokens. Attaching NER information to the respective words did not improve the performance in most cases but still performed better than replacement.

These results contradict the results obtained in previous research [16], which noticed that NER helped most of the times for cyberbullying (CB) detection in Japanese. This could come from the fact that CB is differently realized in English and Japanese. In Japan, revealing victim's personal information, or "doxxing" is known to be one of the most often used forms of bullying, thus NER, which can pin-point information such as address or phone number often help in classification, while this is not the case for English.

Filtering out non-alphabetic characters also reduced the classifier performance most of the time and also got a high score with only one classifier, kNN, which was the weakest classifier overall. Non-alphabetic tokens, which include e.g., punctuation marks, ellipsis, question- of exclamation marks, seem to carry useful information, at least in the context of cyberbullying detection, as removing them reduced the performance comparing to plain tokens due to information loss.

Trying to generalize the feature set ended up lowering the results in most cases with the exception of the very high scores of stopword filtering using traditional classifiers. This would mean that the stopword filtering sometimes succeeded in

removing noise and outliers from the dataset while other generalization methods ended up cutting useful information. Adding information to tokens could be useful in some scenarios as was shown with parts-of-speech tags and using dependency information with CNNs, although using NER was not so successful. Any kind of generalization attempt resulted in a lower performance with CNNs, which shows their ability to assemble more complex patterns from tokens and relations that are unusable by other classifiers.

3.3.3.2 Japanese Cyberbullying Dataset

With the Japanese dataset, most classifiers scored the highest using lemmas, with tokens being not far behind. MLP and CNNs also performed decently on the dependency-based preprocessings. Again, chunking got low performance overall and got outperformed by dependency-based features in MLP and CNNs. The results further confirm that parts-of-speech tags alone are not viable features.

Using POS information as a supplement showed the highest positive impact on the results, increasing the performance most of the time, which can be seen from Table 3.10. Majority of the classifiers had their best result include POS. Adding POS information to the word itself seemed slightly more effective than using separate POS tags also with this dataset.

The rest of the preprocessing types showed mixed results depending on the classifier. Most notably, NER seemed to generally slightly increase the performance of Neural Network and tree based models while struggling with other models like Logistic Regression or SVM. These results are in line with previous research [16]. The effect of stopword filtering and alphabetic filtering was minor, usually only changing the results very little for better or worse depending on the preprocessing type.

Comparing to the English Cyberbullying Dataset, lemmas performed slightly better than tokens and the effectiveness of POS and NER were higher in the Japanese dataset. Whereas stopword filtering performed worse. What makes lemmas slightly better for Japanese could be due to Japanese being an agglutinative language. This increases the base token count, and thus feature density, in comparison to English. Slightly decreasing the number of distinct tokens by lemmatizing could be the key to achieving the optimal amount of information for classification of cyberbullying

in Japanese.

The use of stopwords has been questioned with the Japanese language and even though some lists exist [108], it is much a more difficult preprocessing technique to use when comparing to English, where the use of stopwords is well established. This is because in Japanese each word/morpheme tends to contain useful information, whether syntactic or semantic, unlike in English, where it is easier to filter out words that do not affect the overall meaning of a sentence. The effect of alphabetic filtering was slightly better, yet it still performed average at best. Again, generalizing the dataset seemed to have an adverse effect on the performance most of the time.

3.3.3.3 Polish Cyberbullying Dataset

Pure lemmas worked the best for Polish almost without exception. The performance was clearly superior compared to using tokens. This is completely different from the other datasets where tokens and lemmas only had slight differences in scores. Dependency based features performed very poorly for Polish language, even with Neural Networks.

What makes lemmas significantly better for Polish is most likely due to linguistic differences. Polish is a complex declinative language with a sophisticated morphology and grammar. This increases the base distinct token count and thus feature density. The base feature density for tokens is too high for classifiers to correctly generalize on the dataset of this size. Lemmatizing, which puts all variously declensed and conjugated words back into their dictionary forms, cuts the base feature density by about half, decreasing the complexity of the dataset, and thus greatly increasing the achieved score.

Similarly to the English dataset, stopword filtering performed well compared to other preprocessing techniques, which can be seen from Table 3.13 as it was used in the almost all of the highest scores. The difference being, that the scores were not fluctuating, and the plots (Figure 3.3) for stopwords look similar to the Japanese POS performance. Stopword filtering was clearly the most effective preprocessing method here and clearly increased the scores overall.

Other preprocessing methods had mixed results, slightly increasing or decreasing the results depending on the classifier and preprocessing type. This also includes POS, which clearly had a positive impact in both previous datasets. The

performance of NER and alphabetic filtering were similar to the English dataset.

3.3.3.4 Verification Dataset

Similarly to the English cyberbullying dataset, it can be seen that most of the traditional classifiers scored highest on pure tokens. The difference being that Neural Network based classifiers performed exceptionally well on chunks, as seen in Table 3.15. This might have something to do with the dataset size, as it is around twenty times larger compared to the others. Using lemmas generally got slightly lower scores than tokens, similarly to the English CB dataset. Also, dependency based preprocessings performed rather poorly and I was unable to confirm their effectiveness with CNNs given the larger dataset. However, the supplementary preprocessing methods were shown to be very ineffective and had next to no impact on the results. This is completely contrary to the other datasets as there was at least one supplementary preprocessing method that clearly increased the performance for all other datasets. The reason for this could be the simplicity of the learning problem. Most of the classifiers already scored very high, probably close to the maximum achievable score and thus the effects of extra preprocessing were mostly diminished. The effects could be more visible given a more difficult learning problem.

3.3.4 Classifier Stability

In order to find the best feature sets, I calculated the standard deviations of all cross-validation folds for the English and Japanese datasets. The aim is to find the high performing classifier-preprocessing pairs that are also stable, meaning their performance should have as low variance as possible within the cross-validation folds. I hypothesise that the well performing, stable classifier-preprocessing pairs would be similar across the datasets and could allow even more efficient model training. F-scores and standard errors for each pair are shown in Tables A.1 and A.2 in A.

I calculated the stability scores for Japanese and English datasets by substracting the standard error from the F-scores for each classifier-preprocessing pair. The results are shown in Tables 3.16 and 3.17. For the English dataset, the classifier-

Table 3.16: English dataset: Stability score for classifier-preprocessing pairs

	LBFGS LR	Newton LR	Linear SVM	SGD SVM	KNN	NaiveBayes	RandomForest	AdaBoost	XGBoost	MLP	CNN1	CNN2	Avg for pp
CHNK	0.493	0.484	0.461	0.499	0.18	0.427	0.247	0.351	0.378	0.414	0.401	0.388	0.394
CHNKALPHA	0.422	0.422	0.403	0.42	0.268	0.38	0.323	0.314	0.385	0.31	0.25	0.154	0.338
CHNKNER	0.48	0.487	0.473	0.503	0.197	0.421	0.224	0.331	0.381	0.404	0.458	0.355	0.393
CHNKNERALPHA	0.413	0.416	0.392	0.396	0.258	0.379	0.282	0.308	0.345	0.294	0.275	0.188	0.329
CHNKNERR	0.431	0.437	0.437	0.442	0.2	0.398	0.235	0.298	0.337	0.384	0.383	0.374	0.363
CHNKNERRALPHA	0.391	0.395	0.374	0.381	0.268	0.359	0.273	0.287	0.351	0.262	0.278	0.163	0.315
CHNKNERRSTOP	0.421	0.417	0.41	0.43	0.199	0.37	0.31	0.313	0.352	0.328	0.138	0.121	0.317
CHNKNERRSTOPALPHA	0.338	0.335	0.303	0.32	0.171	0.322	0.316	0.27	0.35	0.178	0.137	-0.03	0.251
CHNKNERSTOP	0.483	0.483	0.461	0.483	0.193	0.412	0.296	0.361	0.394	0.424	0.301	0.103	0.366
CHNKNERSTOPALPHA	0.405	0.396	0.365	0.408	0.168	0.347	0.358	0.324	0.365	0.204	0.091	0.12	0.296
CHNKSTOP	0.481	0.482	0.455	0.484	0.198	0.423	0.359	0.358	0.4	0.437	0.129	0.146	0.363
CHNKSTOPALPHA	0.352	0.363	0.32	0.342	0.169	0.337	0.344	0.364	0.379	0.197	0.178	0.057	0.284
DEP	0.263	0.27	0.17	0.206	0.149	0.337	0.101	0.222	0.249	0.275	0.468	0.445	0.263
DEPALPHA	0.26	0.266	0.216	0.245	0.158	0.312	0.129	0.2	0.258	0.206	0.21	0.202	0.222
DEPNER	0.285	0.28	0.183	0.206	0.15	0.332	0.085	0.214	0.239	0.267	0.475	0.429	0.262
DEPNERALPHA	0.277	0.278	0.244	0.262	0.144	0.325	0.194	0.21	0.265	0.171	0.255	0.216	0.237
DEPNERR	0.26	0.273	0.182	0.206	0.147	0.328	0.107	0.212	0.239	0.261	0.443	0.396	0.255
DEPNERRALPHA	0.254	0.25	0.222	0.243	0.156	0.303	0.119	0.207	0.241	0.153	0.274	0.207	0.219
DEPNERRSTOP	0.244	0.239	0.169	0.175	0.148	0.324	0.122	0.222	0.246	-	0.501	0.453	0.258
DEPNERRSTOPALPHA	0.218	0.219	0.174	0.205	0.163	0.286	0.123	0.204	0.234	0.125	0.255	0.175	0.198
DEPNERSTOP	0.263	0.25	0.167	0.19	0.155	0.32	0.089	0.203	0.244	0.236	0.506	0.46	0.257
DEPNERSTOPALPHA	0.222	0.217	0.204	0.213	0.146	0.256	0.191	0.183	0.23	0.099	0.309	0.223	0.208
DEPSTOP	0.241	0.228	0.162	0.172	0.154	0.328	0.118	0.227	0.245	0.254	0.532	0.377	0.253
DEPSTOPALPHA	0.215	0.222	0.178	0.201	0.167	0.285	0.124	0.205	0.23	0.132	0.277	0.239	0.206
LEM	0.601	0.599	0.594	0.608	0.308	0.494	0.471	0.498	0.519	0.528	0.477	0.4	0.508
LEMALPHA	0.56	0.566	0.541	0.567	0.115	0.478	0.474	0.463	0.526	0.468	0.402	0.368	0.461
LEMNER	0.597	0.599	0.605	0.616	0.301	0.486	0.455	0.483	0.515	0.506	0.49	0.345	0.5
LEMNERALPHA	0.566	0.568	0.562	0.573	0.322	0.472	0.455	0.46	0.504	0.469	0.459	0.387	0.483
LEMNERR	0.523	0.518	0.525	0.526	0.266	0.458	0.417	0.423	0.478	0.463	0.455	0.432	0.457
LEMNERRALPHA	0.507	0.506	0.498	0.503	0.293	0.448	0.417	0.42	0.466	0.473	0.368	0.363	0.439
LEMNERRSTOP	0.517	0.513	0.498	0.512	0.28	0.447	0.482	0.432	0.478	0.398	0.065	0.15	0.398
LEMNERRSTOPALPHA	0.51	0.507	0.478	0.5	0.301	0.435	0.474	0.427	0.457	0.427	0.158	0.081	0.396
LEMNERSTOP	0.597	0.596	0.596	0.613	0.304	0.481	0.515	0.494	0.52	0.482	0.231	0.163	0.466
LEMNERSTOPALPHA	0.575	0.572	0.551	0.567	0.336	0.467	0.508	0.479	0.505	0.425	0.25	0.198	0.453
LEMPOS	0.574	0.58	0.586	0.588	0.152	0.489	0.353	0.463	0.497	0.532	0.471	0.376	0.472
LEMPOSALPHA	0.564	0.562	0.561	0.559	0.312	0.472	0.356	0.441	0.475	0.435	0.451	0.445	0.469
LEMPOSS	0.562	0.558	0.564	0.565	0.205	0.486	0.388	0.428	0.468	0.438	0.4	0.465	0.461
LEMPOSSALPHA	0.56	0.554	0.54	0.549	0.152	0.475	0.41	0.408	0.463	0.45	0.386	0.402	0.446
LEMPOSSSTOP	0.574	0.579	0.581	0.597	0.27	0.49	0.379	0.466	0.49	0.482	0.462	0.491	0.488
LEMPOSSSTOPALPHA	0.564	0.573	0.554	0.575	0.193	0.475	0.399	0.445	0.476	0.461	0.33	0.396	0.453
LEMPOSSTOP	0.593	0.59	0.605	0.602	0.357	0.486	0.447	0.489	0.504	0.43	0.455	0.437	0.5
LEMPOSSTOPALPHA	0.577	0.574	0.564	0.577	0.387	0.469	0.462	0.481	0.491	0.484	0.404	0.471	0.495
LEMSTOP	0.604	0.601	0.592	0.61	0.321	0.492	0.525	0.504	0.533	0.468	0.206	0.122	0.465
LEMSTOPALPHA	0.582	0.574	0.57	0.58	0.154	0.475	0.527	0.476	0.525	0.437	0.165	0.119	0.432
POSS	0.196	0.197	0.195	0.199	0.154	0.192	0.164	0.179	0.176	0.007	0.059	0.055	0.148
POSSALPHA	0.198	0.196	0.194	0.208	0.169	0.192	0.156	0.187	0.168	0.012	0.057	0.053	0.149
POSSSTOP	0.19	0.19	0.189	0.184	0.129	0.189	0.152	0.179	0.164	0.003	-0.05	-0.047	0.123
POSSSTOPALPHA	0.186	0.186	0.183	0.185	0.11	0.181	0.149	0.181	0.157	0.01	-0.053	0.003	0.123
TOK	0.611	0.606	0.611	0.623	0.295	0.495	0.439	0.494	0.525	0.498	0.468	0.403	0.506
TOKALPHA	0.571	0.575	0.555	0.582	0.096	0.481	0.46	0.459	0.518	0.474	0.447	0.342	0.463
TOKNER	0.604	0.606	0.606	0.606	0.265	0.483	0.432	0.483	0.519	0.496	0.489	0.44	0.502
TOKNERALPHA	0.574	0.577	0.563	0.587	0.303	0.466	0.431	0.467	0.517	0.466	0.447	0.391	0.482
TOKNERR	0.526	0.526	0.518	0.527	0.263	0.465	0.407	0.408	0.473	0.458	0.455	0.356	0.449
TOKNERRALPHA	0.519	0.515	0.507	0.513	0.304	0.448	0.401	0.41	0.447	0.441	0.411	0.385	0.442
TOKNERRSTOP	0.516	0.514	0.504	0.509	0.276	0.441	0.486	0.418	0.457	0.448	0.194	0.092	0.405
TOKNERRSTOPALPHA	0.503	0.509	0.49	0.494	0.304	0.432	0.472	0.42	0.444	0.387	0.19	0.164	0.401
TOKNERSTOP	0.599	0.605	0.604	0.609	0.311	0.475	0.502	0.491	0.521	0.465	0.239	0.158	0.465
TOKNERSTOPALPHA	0.576	0.58	0.554	0.581	0.337	0.458	0.501	0.463	0.514	0.441	0.109	0.162	0.44
TOKPOS	0.582	0.584	0.588	0.605	0.187	0.491	0.343	0.45	0.484	0.443	0.425	0.478	0.472
TOKPOSALPHA	0.572	0.569	0.564	0.576	0.385	0.476	0.349	0.428	0.469	0.455	0.365	0.448	0.471
TOKPOSS	0.563	0.568	0.566	0.588	0.177	0.491	0.38	0.417	0.466	0.441	0.482	0.498	0.47
TOKPOSSALPHA	0.57	0.565	0.567	0.575	0.101	0.481	0.404	0.415	0.468	0.351	0.374	0.383	0.438
TOKPOSSSTOP	0.579	0.581	0.585	0.599	0.256	0.481	0.386	0.44	0.495	0.417	0.47	0.495	0.482
TOKPOSSSTOPALPHA	0.582	0.58	0.585	0.585	0.369	0.471	0.404	0.471	0.497	0.451	0.415	0.46	0.489
TOKPOSSTOP	0.599	0.597	0.612	0.613	0.359	0.475	0.449	0.503	0.507	0.451	0.506	0.465	0.511
TOKPOSSTOPALPHA	-	-	-	-	-	-	-	-	-	-	-	-	-
TOKSTOP	0.613	0.608	0.596	0.614	0.317	0.486	0.542	0.509	0.531	0.419	0.156	0.114	0.459
TOKSTOPALPHA	0.587	0.586	0.567	0.584	0.148	0.472	0.528	0.483	0.529	0.428	0.237	0.215	0.447
Avg for clf	0.463	0.463	0.445	0.459	0.228	0.411	0.342	0.372	0.407	0.36	0.321	0.285	

Table 3.17: Japanese dataset: Stability score for classifier-preprocessing pairs

	LBFGS LR	Newton LR	Linear SVM	SGD SVM	KNN	NaiveBayes	RandomForest	AdaBoost	XGBoost	MLP	CNN1	CNN2	Avg for pp
CHNK	0.67	0.663	0.677	0.668	0.289	0.72	0.623	0.47	0.541	0.732	0.745	0.609	0.617
CHNKALPHA	0.686	0.69	0.678	0.662	0.331	0.713	0.641	0.47	0.533	0.715	0.718	0.6	0.62
CHNKNER	0.716	0.721	0.735	0.732	0.349	0.741	0.699	0.639	0.673	0.777	0.75	-	0.685
CHNKNERALPHA	0.72	0.725	0.726	0.717	0.396	0.738	0.685	0.627	0.669	0.738	0.743	-	0.68
CHNKNERR	0.722	0.721	0.724	0.707	0.366	0.727	0.683	0.631	0.667	0.752	0.766	0.667	0.678
CHNKNERRALPHA	0.687	0.696	0.709	0.7	0.366	0.722	0.678	0.615	0.649	0.717	0.743	0.632	0.66
CHNKNERRSTOP	0.709	0.704	0.727	0.699	0.359	0.699	0.687	0.637	0.651	0.758	0.749	0.642	0.668
CHNKNERRSTOPALPHA	0.689	0.682	0.705	0.686	0.379	0.704	0.665	0.607	0.628	0.708	0.695	0.63	0.648
CHNKNERSTOP	0.714	0.715	0.731	0.752	0.383	0.738	0.699	0.643	0.662	0.757	0.766	-	0.687
CHNKNERSTOPALPHA	0.721	0.723	0.728	0.734	0.357	0.728	0.676	0.639	0.659	0.715	0.733	-	0.674
CHNKSTOP	0.661	0.661	0.674	0.671	0.306	0.682	0.608	0.457	0.537	0.73	0.716	0.667	0.614
CHNKSTOPALPHA	0.646	0.651	0.666	0.651	0.327	0.687	0.595	0.439	0.539	0.674	0.699	0.571	0.595
DEP	0.609	0.611	0.619	0.625	0.307	0.621	0.484	0.358	0.456	0.737	0.736	0.546	0.559
DEPALPHA	0.595	0.594	0.614	0.609	0.27	0.613	0.456	0.35	0.444	0.723	0.686	0.588	0.545
DEPNER	0.674	0.682	0.707	0.713	0.331	0.684	0.649	0.625	0.658	0.765	0.727	-	0.656
DEPNERALPHA	0.684	0.683	0.709	0.712	0.179	0.675	0.642	0.631	0.658	0.766	0.73	-	0.643
DEPNERR	0.612	0.617	0.615	0.625	0.279	0.616	0.486	0.336	0.45	0.756	0.747	0.553	0.558
DEPNERRALPHA	0.593	0.594	0.609	0.604	0.274	0.615	0.448	0.342	0.448	0.735	0.735	0.553	0.546
DEPNERRSTOP	0.575	0.569	0.573	0.567	0.343	0.57	0.419	0.324	0.397	0.765	0.736	0.445	0.524
DEPNERRSTOPALPHA	0.509	0.508	0.51	0.516	0.326	0.528	0.354	0.309	0.348	0.73	0.698	0.572	0.492
DEPNERSTOP	0.662	0.663	0.681	0.689	0.414	0.66	0.631	0.611	0.628	0.776	0.727	-	0.649
DEPNERSTOPALPHA	0.665	0.661	0.683	0.672	0.211	0.699	0.632	0.617	0.639	0.757	0.719	-	0.632
DEPSTOP	0.572	0.579	0.576	0.577	0.341	0.573	0.427	0.34	0.405	0.748	0.703	0.619	0.538
DEPSTOPALPHA	0.518	0.518	0.522	0.528	0.33	0.535	0.369	0.327	0.363	0.718	0.716	0.579	0.502
LEM	0.783	0.777	0.8	0.776	0.477	0.803	0.739	0.589	0.683	0.846	0.837	0.692	0.734
LEMALPHA	0.779	0.776	0.8	0.785	0.472	0.793	0.728	0.586	0.682	0.839	0.831	0.784	0.738
LEMNER	0.778	0.769	0.795	0.794	0.523	0.794	0.768	0.705	0.74	0.853	0.845	0.595	0.747
LEMNERALPHA	0.781	0.785	0.807	0.809	0.51	0.798	0.771	0.697	0.746	0.838	0.829	0.744	0.76
LEMNERR	0.765	0.758	0.777	0.767	0.468	0.786	0.752	0.7	0.726	0.825	0.79	0.632	0.729
LEMNERRALPHA	0.77	0.775	0.778	0.772	0.471	0.787	0.749	0.708	0.741	0.816	0.81	0.686	0.739
LEMNERRSTOP	0.763	0.756	0.77	0.769	0.469	0.769	0.772	0.703	0.717	0.82	0.82	0.791	0.743
LEMNERRSTOPALPHA	0.766	0.764	0.772	0.768	0.457	0.769	0.757	0.702	0.735	0.788	0.8	0.75	0.736
LEMNERSTOP	0.769	0.773	0.791	0.802	0.533	0.805	0.774	0.71	0.731	0.856	0.858	0.784	0.766
LEMNERSTOPALPHA	0.78	0.779	0.806	0.806	0.543	0.82	0.783	0.706	0.729	0.826	0.815	0.71	0.759
LEMPOS	0.785	0.786	0.83	0.827	0.677	0.816	0.762	0.691	0.742	0.847	0.845	0.595	0.767
LEMPOSALPHA	0.781	0.768	0.821	0.827	0.674	0.802	0.741	0.675	0.739	0.839	0.845	0.647	0.763
LEMPOSS	0.792	0.8	0.83	0.825	0.533	0.83	0.752	0.671	0.747	0.847	-	0.755	0.762
LEMPOSSALPHA	0.805	0.793	0.83	0.818	0.699	0.827	0.731	0.626	0.718	0.832	0.843	0.544	0.756
LEMPOSSSTOP	0.791	0.79	0.832	0.829	0.491	0.833	0.743	0.688	0.742	0.845	0.856	0.796	0.77
LEMPOSSSTOPALPHA	0.783	0.792	0.816	0.813	0.685	0.822	0.745	0.661	0.713	0.835	0.822	0.673	0.763
LEMPOSSTOP	0.779	0.766	0.826	0.824	0.423	0.815	0.74	0.696	0.751	0.846	0.85	0.783	0.758
LEMPOSSTOPALPHA	0.774	0.765	0.804	0.817	0.672	0.815	0.733	0.664	0.733	0.817	0.832	0.752	0.765
LEMSTOP	0.771	0.775	0.794	0.778	0.482	0.78	0.755	0.577	0.669	0.84	0.847	0.792	0.738
LEMSTOPALPHA	0.772	0.774	0.788	0.776	0.487	0.796	0.744	0.587	0.668	0.817	0.823	0.774	0.734
POSS	0.607	0.616	0.595	0.581	0.507	0.585	0.604	0.614	0.608	0.621	0.638	0.458	0.586
POSSALPHA	0.61	0.611	0.584	0.552	0.502	0.586	0.606	0.616	0.611	0.613	0.624	0.537	0.588
POSSSTOP	0.607	0.627	0.604	0.587	0.47	0.588	0.616	0.615	0.599	0.607	0.599	0.462	0.582
POSSSTOPALPHA	0.581	0.582	0.54	0.558	0.381	0.569	0.575	0.576	0.562	0.567	0.537	0.455	0.54
TOK	0.777	0.775	0.79	0.785	0.448	0.795	0.737	0.573	0.673	0.845	0.833	0.779	0.734
TOKALPHA	0.781	0.774	0.788	0.784	0.449	0.798	0.73	0.578	0.66	0.827	0.834	0.78	0.732
TOKNER	0.786	0.785	0.812	0.808	0.6	0.817	0.732	0.63	0.699	0.828	0.825	0.662	0.749
TOKNERALPHA	0.773	0.779	0.801	0.794	0.493	0.807	0.762	0.702	0.724	0.846	0.842	0.804	0.761
TOKNERR	0.758	0.756	0.78	0.783	0.463	0.778	0.762	0.692	0.716	0.81	0.828	0.771	0.741
TOKNERRALPHA	0.769	0.765	0.778	0.768	0.441	0.778	0.749	0.703	0.725	0.815	0.803	0.696	0.733
TOKNERRSTOP	0.744	0.746	0.767	0.762	0.453	0.764	0.751	0.688	0.722	0.815	0.829	0.682	0.727
TOKNERRSTOPALPHA	0.761	0.752	0.774	0.77	0.444	0.751	0.749	0.696	0.732	0.802	0.794	0.723	0.729
TOKNERSTOP	0.777	0.774	0.803	0.798	0.494	0.808	0.759	0.7	0.735	0.827	0.83	0.792	0.758
TOKNERSTOPALPHA	0.764	0.763	0.784	0.805	0.5	0.801	0.766	0.696	0.726	0.85	0.848	0.808	0.759
TOKPOS	0.779	0.777	0.801	0.812	0.607	0.813	0.767	0.688	0.727	0.821	0.825	0.742	0.763
TOKPOSALPHA	0.793	0.785	0.819	0.83	0.665	0.81	0.737	0.668	0.74	0.855	0.846	0.655	0.767
TOKPOSS	0.79	0.798	0.821	0.812	0.504	0.831	0.75	0.662	0.725	0.845	-	0.662	0.745
TOKPOSSALPHA	0.756	0.761	0.807	0.819	0.694	0.808	0.707	0.647	0.7	0.845	0.845	0.632	0.752
TOKPOSSSTOP	0.798	0.79	0.807	0.814	0.677	0.826	0.738	0.676	0.739	0.833	0.8	0.621	0.76
TOKPOSSSTOPALPHA	0.794	0.803	0.816	0.822	0.477	0.826	0.728	0.674	0.743	0.845	0.863	0.74	0.761
TOKPOSSTOP	0.764	0.773	0.82	0.821	0.557	0.803	0.739	0.685	0.738	0.825	0.835	0.601	0.747
TOKPOSSTOPALPHA	-	0.765	0.813	0.821	0.527	0.817	0.732	0.676	0.73	-	-	-	0.735
TOKSTOP	0.776	0.774	0.787	0.778	0.451	0.786	0.737	0.568	0.662	0.844	0.848	0.759	0.731
TOKSTOPALPHA	0.774	0.768	0.787	0.794	0.459	0.802	0.752	0.57	0.659	0.811	0.797	0.66	0.719
Avg for clf	0.721	0.721	0.739	0.736	0.453	0.741	0.68	0.603	0.653	0.785	0.779	0.665	

preprocessing pairs that got the highest stability scores are Logistic Regressions and SVMs with tokens and lemmas with parts of speech and their derivatives, highest being tokens with parts of speech and stopword filtering (TOKPOSSTOP).

The same preprocessing types, tokens and lemmas with parts of speech and their derivatives, also had high scores for Japanese with logistic regressions and SVMs, although the actual highest performing classifiers were Neural Networks (CNN1, MLP) instead. Tokens and lemmas with named entity recognition also scored high. Based on these results one can say that parts of speech information added to tokens or lemmas should be the preferred feature sets when training as they increase the performance of the most effective classifiers compared to plain tokens or lemmas.

3.4 General Discussion

Overall, the highest performing classifiers were one-layer CNN for Japanese and Verification datasets, SVM for English and Logistic Regression for the Polish dataset. The baseline classifiers, KNN and Naive Bayes, and tree based methods, Random forest, AdaBoost, XGBoost, had mediocre results at best. Using only POS tags as features had extremely low performance with all of the datasets, so they were excluded from the analysis.

An interesting discovery is that using plain tokens rarely had the best performance out of the proposed feature sets as can be seen from Tables 3.8, 3.10, 3.13 and 3.15. This proves the effectiveness of linguistics-based feature engineering instead of using just words as features. For example, the one-layer CNN's performance with English increased from 0.659 (TOK) F-score to 0.741 (DEPSTOP), indicating that information about structure seems important.

The fact that feature sets with linguistic information managed to outperform the use of tokens clearly shows their potential in feature engineering. for some models, like the one-layer CNN with the English dataset, the increase was truly significant, almost 0.1 F-score using linguistic embeddings. I also discovered that adding parts of speech information to tokens or lemmas produces the most stable and high performance feature sets and they should be preferred over plain tokens or lemmas. This discovery also raises a question: how would linguistic preprocessing

affect state-of-the-art pretrained models? For example, RoBERTa [121] fine tuned on the English dataset shows an F-score of 0.797, which is similar to the highest scores by other models. Actually, the best score by SGD SVM is 0.798 which is slightly higher.

It is interesting that a simple method like SVM can outperform a complex modern text classifier when using the right feature set. This shows that they should not be underestimated as with correct preparations, they can reach similar performance to state-of-the art models with much less computational power required. Perhaps the performance of RoBERTa could also be increased by feature engineering and applying embeddings with linguistic information. This needs to be explored further in future research as there is clearly a potential point of improvement.

3.4.1 Feature Density

In general, most of the classifier performances, with the exception of CNNs, had a strong negative correlation with FD. So these classifiers seem to have a weaker performance if a lot of linguistic information is added. These results suggest that for non-CNN classifiers there is no need to consider preprocessings with a high FD, such as chunking or dependencies, as they had a considerably lower performance. The best performance was in most cases between 50-200% of the base FD (TOK) depending on the dataset and classifier.

For CNNs I was expecting a positive correlation with FD, at least with the Japanese dataset, considering the results of the previous research [16]. However, the results were different and I was unable confirm the positive correlation between classifier performance and FD for the Japanese dataset. Still, in some cases such as the English dataset, there was a very weak positive or no correlation between FD and the classifier performance, with the higher FD datasets performing equally or even slightly better. This could mean that there is potential in the higher FD preprocessing types, namely, dependencies for CNNs. As the FD that had the best performance varied slightly throughout the classifiers and language, more exact ideal feature densities need to be confirmed for each classifier using datasets of different sizes and fields to make as accurate ranking of classifiers by FD as possible.

For the English dataset, the best results were usually obtained by slightly increasing the base FD, while for the Japanese and Polish dataset decreasing the

base density tended to work better. This could be because Japanese and especially Polish are a lot more complex languages compared to English. This is also indicated by the fact that the LD of Japanese and Polish were higher than English.

The clearly visible optimal range of Feature Densities usually held between 50 to 70% of the proposed feature sets, which means that by ignoring the redundant preprocessings, the total training time could be reduced. Preprocessings using Chunking and Dependencies were usually left outside of the range which means that they add too much information for the classifiers to handle. Neural Networks make an exception here with some of the datasets.

I found out that the optimal feature densities tend to lie in specific ranges for all of the datasets. Even though the ranges were different for each classifier, it is still possible to use this information to further decrease the required amount of work. Instead of running all of the experiments at once, I propose first discarding the FD ranges where the overall weakest feature sets according to the characteristics of the classifiers, POS, CHNK for all, and DEP for other than neural classifiers. Then running a subset of the experiments using the generally most effective feature sets, TOK/LEM+POS and variants and also DEP for neural classifiers. Afterwards, it is possible to run more experiments with similar feature densities as the current peak to find the maximum performance.

Let us take the Polish dataset with SGD SVM classifier as an example. First, I discard the feature sets that assumed to have a weak performance (POS, CHNK, DEP). Then train the classifier on the datasets found the most effective with English and Japanese datasets (TOK/LEM+POS and variants). At this point I have run 16 of the 68 total experiments and gotten an F-score of 0.498 (LEMPOSSTOP), which is already considerably higher than TOK (0.456). If not satisfied with the current high score here, the solution is to start running more experiments with similar FD to LEMPOSSTOP. This way it is possible find a slightly higher score of 0.500 (LEMSTOP), but this would require eight additional experiments if starting from the FDs closest to LEMPOSSTOP. So, using this method, the classifier's performance can be considerably improved with less than a quarter of the original amount of experiments. I assume that the method could be also applied to other classifiers than those experimented with in this study, including modern models.

3.4.2 Dataset Complexity

From the classification results it can be seen that the dataset with the lowest Lexical Density, the verification dataset, was the easiest to classify. While the highest Lexical Density dataset (Polish) was the hardest. However, the Japanese dataset got higher F-scores compared to English even though it had a higher Lexical Density. This means that natural language dataset complexity cannot be measured by using Lexical Density alone, even if it could be used to measure the complexity of the language itself.

There are some interesting points to note however when considering Feature Density. The relative change in Feature Density when lemmatizing each dataset matches the ranking of the dataset scores. This also applies to dependency patterns, except in the case of the Japanese dataset, which actually becomes the most dense, despite achieving the highest scores. From the results one can see that there are huge differences in classification results even when given datasets of a similar size and topic. This means it is not enough to develop the tools and models with just English in mind. Even though most of the research in NLP is done in English, it is a much less complex language when compared to other more morphologically rich languages, thus state of the art results achieved for English are not representative globally for the task in question, but rather locally, for the task done in this particular language. Additionally, even if it cannot be perfectly quantified by a simple measure like Feature Density, the complexity of languages affects the dataset complexity which in return impact the results of the classifiers applied. Even if a model has a high score in English it does not mean it excels with other languages.

3.4.3 Linguistic Preprocessing

Using token-based preprocessings worked slightly better when classifying the English and Verification datasets. For the Japanese dataset, lemmas performed slightly better compared to tokens and for the Polish dataset, lemmas were clearly superior when compared to using tokens. What causes lemmas to work better for Japanese and Polish is most likely due to linguistic differences. Japanese and Polish are more complex languages than English, which can also be seen as an increase in the base distinct token count and thus feature density. The base feature density for

tokens is too high for classifiers to correctly generalize on the dataset of this size. Lemmatizing, which puts all variously declensed and conjugated words back into their dictionary forms, cuts the base feature density, decreasing the complexity of the dataset, and thus increasing the achieved score. This can be seen especially well the case of Polish when the base FD is cut almost in half by lemmatizing, which greatly increases the score.

Chunking and dependency based preprocessings did not perform well compared to tokens or lemmas overall. They showed poor performance with all classifiers and datasets. Only exceptions being with English and Japanese datasets, where dependency based preprocessings showed some potential with the Neural Networks, although still outperformed by tokens or lemmas.

Stopword filtering was to be the one of the most effective preprocessing techniques for both English and Polish. Although for English, the results sometimes fluctuated. For Japanese, stopwords were not effective. This is because in Japanese each word/morpheme tends to contain useful information. This is different from English, where it is easier to filter out words that do not affect the overall meaning of a sentence.

Using parts-of-speech tags was shown to be very effective with both English and especially with Japanese, where it was clearly the most effective preprocessing method. Merging parts-of-speech information to the words themselves usually achieved a higher score than using them separately as features. This keeps the information directly connected to the word itself, which seems a better option from the point of view of information preservation.

Using Named Entity Recognition information had mixed results, reducing or only slightly increasing the classifier performance with English and Polish datasets. Only giving generally positive results with Japanese dataset. Overall, the performance of using NER seemed clearly inferior compared to stopword filtering or parts-of-speech information. Replacing words with their NER information seems to cause too much information loss and reduces the performance, while attaching NER information to the respective words performed somewhat better.

Filtering out non-alphabetic characters had the poorest performance of all the supplementary preprocessing types and reduced the classifier performance most of the time. Non-alphabetic tokens, which include e.g., punctuation marks, ellipsis,

question- of exclamation marks, seem to carry useful information, at least in the context of cyberbullying detection, as removing them reduced the performance comparing to plain tokens due to information loss, which should be taken into account in future research.

In most cases, preprocessing had a considerable positive effect on the scores. I was able to find preprocessing methods that outperformed the base method of using pure tokens most of the time, which shows the potential of using linguistic information as a method for increasing classifier performance. Based on the experiments, it is suggested to omit pure POS based and CHNK based feature sets for all classifiers due to their overall poor performance. Also, dependency based feature sets should be omitted for non-neural classifiers.

Only in the case of the verification dataset, which was found to be most likely too simple of a learning problem, the preprocessing methods were shown to be ineffective and had next to no impact on the results. The reason for this could be the simplicity of the learning problem. Most of the classifiers probably scored close to the maximum achievable score and thus the effects of extra preprocessing vanished. In the future, the effects of linguistic preprocessing should be also confirmed with state-of-the-art pretrained language models.

3.4.4 Effect on Cyberbullying Detection

Leaving out the weaker feature sets and classifiers can be used to reduce time and effort when creating cyberbullying detection systems. The savings brought by the method do not only help to protect the environment but also allow quicker and more efficient development of CB detection systems. This is crucial as each day countless number of people become the victims of cyberbullies.

According to the experiments, although there were differences with F-scores for individual preprocessing types, the highest stability score datasets, tokens and lemmas with parts of speech information and their derivatives, were similar for both English and Japanese. The difference being the effectiveness of named entity recognition for Japanese. Also, from the applied classifiers, only SVMs, logistic regression and neural network models should be considered when dealing with CB.

The fact that the CNN's performance increased drastically when using dependency information instead of plain tokens with the English dataset could hint

that structure and syntactic relations between words could be important when classifying CB entries. This needs to be explored further in the future.

Also, SVM (max F1=0.798), MLP (max F1=0.796) and Logistic Regression (max F1=0.793) achieved similar scores to RoBERTa (F1=0.797) on the English dataset, with SGD SVM actually slightly outperforming it with one of the feature sets (TOKPOS). This shows that the method can not only be used to save resources and time, but also to increase the performance of cyberbullying detection models.

3.4.5 Environmental Effect

To see the concrete environmental effect of the method, let us take a look at the English dataset, where the savings in power when training CNN were around 21kWh. According to European Environmental Agency (EEA) [3], the average greenhouse gas emissions from generating electricity was at 275 $g\ CO_2e$/kWh in 2019. This means the emissions generated by the training of CNN could be estimated at around 5.8 $kg\ CO_2e$. For comparison, the average new passenger car in the European Union in 2019, according to EEA, emits around 122 $g\ CO_2e$ per kilometer driven. This means that leaving out the weaker feature sets by utilizing FD could save as much as driving a new car for almost 50 kilometers in greenhouse gas emissions when training a simple CNN model.

Although the effect does not seem that overwhelming, one has to take into account that the models tested here were quite simple. Assuming the hypothesis, if the method would be applied to more resource intensive modern classifiers, the savings would become considerably more significant.

3.5 Additional Experiments with Linguistically-Backed Word Embeddings

Adding to the previous experiments, I trained Word2Vec Skip-Gram embeddings with encoded linguistic information and also by using dependency structure based contexts similarly to Levy and Goldberg [68]. I then evaluated these methods using

[3] `https://www.eea.europa.eu/`

different kinds of neural network models with Support Vector Machines [110] as the baseline.

3.5.1 Setup

In order to train the linguistically-backed embeddings, I first preprocessed the dataset in various ways, similarly to Levy and Goldberg [68] and Ptaszynski et al. [16]. The preprocessing was done using spaCy NLP toolkit (https://spacy.io/). Some of these preprocessing methods are the same as used in the previous experiment.

- **Tokenization:** words separated by spaces (later: TOK).
- **Lemmatization:** like the above but in generic (dictionary) forms of words ("lemmas") (later: LEM).
- **Encoded parts of speech:** parts of speech information is merged with LEM or TOK (later: POS).
- **Encoded dependency structures:** token pairs with syntactic relations encoded between them (later: DEP).
- **Dependency-based contexts:** The use of dependency relations instead of a fixed window of tokens as context when training embeddings (later: DEPC) [68].

I generated a Word2Vec Skip-Gram language model from each of the processed dataset versions. This resulted in separate models for each of the datasets, Tokens-Skip-Grams, Lemmas-Skip-Grams, Tokens-POS-Skip-Grams, Lemmas-POS-Skip-Grams, DEP-Skip-Grams and DEPC-Skip-Grams.

The dependency-based context embeddings (DEPC-Skip-Grams) are the same dependency embeddings used in the previous research by Levy and Goldberg [68]. Other embeddings used a fixed context window size of 5. The embeddings were pretrained on a 1GB sample of English Wikipedia dataset using Gensim [122] with 300 dimensions.

To evaluate the embeddings, I used a linear Support Vector Machine (SVM) [110] as the baseline. I also used different neural network architectures, Recurrent

Neural Network with Long short-term memory (LSTM), Convolutional Neural Network (CNN) and Multilayer Perceptron (MLP).

The preprocessing provides 7 separate datasets for both the Wikipedia dataset and the target cyberbullying dataset. I trained the embeddings and performed the experiments once for each type of preprocessed dataset. Each of the classifiers (sect. 3.2.3) were tested on each version of the dataset in a 10-fold cross validation procedure. The evaluation results were calculated using balanced F-score. As the dataset was not balanced, it was required to weigh the classes accordingly. I ran two sets of experiments. First pretraining the embeddings on the Wikipedia dataset prior to training the classifiers on the target cyberbullying dataset. Second, training the embeddings *ad hoc* on the target dataset itself.

3.5.2 Evaluation of linguistic embeddings

From the results presented in the upper half of Table 3.18 it can be seen that most of the classifiers scored highest on raw lemmas embeddings, with the exception of MLP. As expected, the baseline SVM model had the lowest scores overall. LSTM had the lowest after the baseline, probably due to the small size of the dataset. CNNs had the highest scores across the board followed closely by MLP. The main difference in the performances of these two classifiers was that CNNs scored much higher on lemmas.

Table 3.18: F-scores of classifier-embedding type pairs.
Pretrained: upper half, Ad hoc: lower half

	TOK	TOK POS	LEM	LEM POS	DEP	TOK[68]	DEPC[68]
SVM	0.481	0.483	0.484	0.483	0.48	0.481	0.497
LSTM	0.506	0.512	0.538	0.53	0.492	0.531	0.527
CNN	0.754	0.712	0.757	0.702	0.654	0.751	0.749
MLP	0.538	0.741	0.656	0.715	0.679	0.752	0.741
SVM	0.793	0.791	0.784	0.788	0.568		
CNN	0.659	0.626	0.67	0.665	0.682		
MLP	0.796	0.787	0.786	0.783	0.594		

Lemmas achieved the highest scores, being slightly better than other embeddings with the exception of MLP, where Levy and Goldberg's [68] token embeddings

and dependency context embeddings were clearly better. The differences in scores between the token embeddings used here and Levy and Goldberg's could be explained by the differences in the training data size as there is a noticeable difference in the vocabulary size of the embeddings used here versus theirs (40,000 vs 180,000). This could suggest that lemmatization can be effective as a technique in increasing the performance of embeddings for a cyberbullying detection task.

The POS embeddings did not score well compared to their simpler counterparts in most cases. The only exception being MLP where they showed a clear performance boost, with TOKPOS scoring especially high. One of the reasons for the generally lower performance could be related to the increased sparsity of the dataset due to adding POS information. This could be corrected by applying a larger dataset for training the word embeddings. The difference could also be in the forming of the embeddings themselves. Because of this, I am planning to conduct qualitative evaluation and manually inspect the embeddings more closely in the future.

The proposed dependency embeddings scored the lowest and were outperformed by all other types of embeddings in basically all cases. One of the reasons could be again related to the greatly increased sparsity of the dataset due to the added dependency information. In the future I plan to train the embeddings on a larger dataset and also conduct qualitative evaluation to study the differences between the implementation used here and Levy and Goldberg's [68].

According to the results, using dependency based embeddings does not offer any noticeable improvements on cyberbullying detection task. However, this needs to be confirmed using a larger training set for the embeddings. The task should also be evaluated using other cyberbullying datasets.

3.5.3 Comparison with *ad hoc* embeddings

To see the effect of using pretrained word embeddings over *ad hoc* embeddings, I also ran the experiments without pretraining the embeddings. For the baseline SVM classifier, I used a tf-idf weighing scheme to produce a BoW language model of the evaluation (cyberbullying) dataset. For the neural network models, I trained the embeddings on the evaluation datasets themselves as part of the networks using Keras' embedding layer with random initial weights.

The results with *ad hoc* embeddings are shown in the lower half of Table 3.18.

First thing to note is that SVM performed significantly better with the BoW language model instead of pretrained embeddings. The reason most likely is that SVM uses the embeddings as they are and no adjustment to the cyberbullying specific vocabulary is done during training of the classifier, whereas the BoW model was trained on the cyberbullying data itself and captures its concepts.

CNN's performance on the other hand was a significantly worse without using pretrained word embeddings. The difference in scores clearly shows why pretrained language models are popular, as the CNN model gains a noticeable boost from a very general dataset completely irrelevant to the target. This also shows the nature of the CNN model earlier observed by Kim [117], that CNNs greatly benefit from pretrained word embeddings. The same does not apply to MLP though as it seemed to perform slightly better without pretraining. With a larger pretraining dataset, the situation could be different.

Chapter 4

Transfer Language Selection for Cyberbullying Detection

In the previous Chapter, I demonstrated how to hasten model development when a proper dataset is available. In this Chapter however, I wanted to know how to deal with the problem of cyberbullying in a language that has no available training data. One of the biggest challenges in NLP is related to the availability of data. This means that in order to successfully train models for all of the world's languages, one would need to annotate a dataset for each language. This is a very difficult and costly task and has led to a small number of high-resource languages to dominate the field, most notably English. This imbalance in the distribution of resources among languages calls for the need to develop solutions that would aid in model development for languages lacking data.

One solution that enables the development of abusive content detection models without the need of going through the annotation process is to leverage knowledge from other languages that have a proper training dataset available. Using Zero-shot cross-lingual transfer learning does not require any labeled data in the target language, meaning that the model is trained completely by using labeled data from other languages. This can also be used as a temporary solution to train models for low-resource languages.

However, the current methods for choosing a language for as the cross-lingual transfer source are mainly based on the individual's own judgement based on their field experience and accumulated theoretical knowledge or simply choosing

languages from the same language family [31]. The problem with the current selection methods are that they are completely unoptimized and prone to bias from the practitioner. In fact, one could argue that there is no systematic method that would give an actual score or ranking for the transfer language candidates.

This Chapter aims to answer the need of developing a method of selecting languages for cross-lingual transfer learning in order to aid in development of cyberbullying detection systems also for languages lacking data. The approach to this problem is to explore, whether different linguistic similarity metrics could be used for finding the optimal candidates for cross-lingual transfer. Supported by the findings of Gaikwad et al. [30], I hypothesize that linguistic similarity correlates with cross-lingual transfer efficacy, meaning that using more similar languages would yield a higher classification score. In practice, I fine tune cross-lingual pretrained language models, specifically mBERT and XLM-R, separately on each of the proposed languages (English, German, Danish, Polish, Russian, Croatian, Japanese, Korean) and then perform zero-shot classification on the rest of the languages of the proposed set.

4.1 Datasets

In order to confirm the hypothesis, I used offensive language (hate speech, cyberbullying) datasets from seven different languages, namely English, German, Danish, Polish, Russian, Japanese and Korean. I chose these languages as they had high quality datasets compared to other options and because the languages represent three different language families (English, German, Danish - Germanic; Polish, Russian - Slavic; Japanese, Korean - Koreano-Japonic language family). This also makes it possible to study the efficacy of transfer learning between and within language family groups. Preliminarily, I had a few datasets of good quality (English, Polish, Japanese). I tried to keep the quality high also for the other languages to the best of my ability. For Germanic languages, there are some good quality datasets available, but then same cannot be said about Russian and Korean and I had to loosen the standards for these two languages. Also, the aim was to use as many cyberbullying datasets as possible, but due to unavailability, I had to settled for other abusive language datasets in many cases. Key statistics of

the applied datasets are shown in Table 4.1. Training and evaluation splits were retained from original datasets if possible, otherwise datasets were split to 80% training and 20% evaluation.

Table 4.1: Statistics of the applied offensive language identification datasets

	ENG	**GER**	**DAN**	**POL**	**RUS**	**JPN**	**KOR**
Category	CB	Offense	Offense	Offense	Toxic	CB	Hate
Samples	12,772	8,407	3,289	34,953	14,412	4,096	189,995
Offensive samples	913	2,838	425	7,367	4,826	2,048	89,999
Non-offensive samples	11,859	5,569	2,864	27,586	9,586	2,048	99,996
Split (train/eval)	80/20	60/40	90/10	83/17	80/20	80/20	80/20

4.1.1 English Dataset

The first dataset for the experiments was the Kaggle Formspring Dataset for Cyberbullying Detection [103]. There was one major problem with the original dataset however, as the original annotations for the data were carried out by untrained laypeople. It has been proven before that the annotations for topics like online harassment and cyberbullying should be done by experts [5]. Therefore, the dataset was re-annotated with the help of experts with sufficient psychological background to assure high quality annotations [104]. In the research I applied the re-annotated version for more accurate results.

The dataset contains approximately 300 thousand of tokens. There was no visible difference in length between the posted questions and answers, both being approximately 12 words long on average. On the contrary, the harmful (CB) entries were usually slightly but insignificantly shorter compared to the non-harmful (non-CB) samples (approx. 23 vs. 25 words). The amount of harmful samples was also substantially smaller compared to the amount of non-harmful samples, around 7% of the whole dataset, which is approximately the same as the real-life amount of profanity encountered on SNS [5].

4.1.2 German Dataset

The German dataset originates from the 2018 GermEval offensive language identification shared task [123] and contains around 8,000 entries collected from Twitter. They decided against collecting a natural sample as it would have ended up making the portion of offensive tweets too small. They also decided against sampling by specific query terms. Instead they heuristically identified users that regularly post offensive tweets and sampled their timeline. This allowed for more offensive tweets in comparison to taking a natural sample without biasing the dataset with specific terms. However this caused certain topics to dominate in the extracted data, like the situation of migrants or the German government. So they decided to bias the data collection by sampling further arbitrary tweets containing common terms found in these topics like names of politicians and the word “refugee".

There are some rules regarding the selection of the tweets put up by the authors. Each tweet is written in German and contains at least five ordinary alphabetic tokens. Also the tweets do not contain any URLs and retweets were not allowed. In splitting the data into training and test sets, the authors decided to assign any given user's complete set of tweets to either the training set or the test set. This way, they could avoid the fact that the classifiers could benefit from learning user-specific information.

4.1.3 Danish Dataset

Sigurbergsson and Derczynski [124] collected the Danish dataset from Facebook and Reddit. The final dataset contains 3600 user-generated comments, 800 from Ekstra Bladet on Facebook, 1400 from r/DANMAG and 1400 from r/Denmark.

After collecting the initial corpus, they published a survey on Reddit in order to maximize the number of user-generated comments belonging to the classes of interest (offensive language), where they asked Danish speaking users to suggest offensive, sexist, and racist terms. This lexicon was then used to find potentially-offensive comments. A subset was then taken from the comments remaining in the corpus to fill the remainder of the final dataset. This helped to ensure that the data would have coverage beyond just the terms found in the lexicon.

They based the annotation procedure on the guidelines and schemas presented

by Zampieri et al. [125]. As a warm-up procedure, the first 100 posts were annotated by two annotators and the results were compared. This exercise was used to refine the understanding of the task and to discuss the mismatches in these annotations. They assessed the similarity of the annotations using a Jaccard index.

4.1.4 Polish Dataset

The Polish corpus is a combination of two datasets. One originates from PolEval workshop from 2019 [106], collected from Twitter discussions. Another one was collected from Wykop[1], which is a Polish social networking service. As feature selection and feature engineering have been proven to be integral parts of cyberbullying detection [16, 107], the entries are provided as such, without additional preprocessing to allow researchers using the datasets apply their own preprocessing methods. The only preprocessing applied to the dataset was done only to mask private information, such as personal information of individuals (usernames, etc.).

The datasets were initially annotated by laypeople, and further corrected by experts in case of disagreements. The laypeople agreed on majority of the annotations. This is mostly due to the fact that the annotators mostly agreed upon non-harmful entries, which take up most of the dataset. When considering the harmful class, the annotators only fully agreed upon less than two percent of the entries. Moreover, some of the fully-agreed entries needed to be corrected to the opposite class in the end by the expert annotator, which shows that using laypeople does not provide accurate enough annotations in the field of offensive language identification. It could be said that layperson annotators can tell with a decent level of confidence that an entry is not harmful (even if it contains some vulgar words), and they can spot, to some extent, if the entry is somehow harmful. Though in most cases they are unable to provide a reasoning for their choice. This provides further proof that for specific problems such as cyberbullying, an expert annotation is required [5].

[1] `https://www.wykop.pl`

4.1.5 Russian Dataset

The Kaggle Russian Language Toxic Comments Dataset [2] is the collection of annotated comments from Russian online communication platforms. 2ch, which is a popular Russian anonymous image board and Pikabu, which could be considered the Russian equivalent of Reddit. The dataset was published on Kaggle in 2019. It consists of a total of 14,412 comments, out of which 4,826 texts are labeled as toxic, and the remaining 9,586 are labeled as non-toxic. The average length of the comments is around 175 characters. The minimum length being 21, and the maximum length being 7403 characters. The annotators of this dataset are unknown so unfortunately I cannot say anything about its quality. Although the annotations were validated with the help of Russian language speakers (laypeople) using a crowd sourcing application [126].

4.1.6 Japanese Dataset

The Japanese cyberbullying dataset I used for the experiments was created by combining two separate datasets. The first of which was originally described by Ptaszynski et al. [9], and also widely used in other research [38, 41, 11, 15, 16]. It contains 1,490 harmful and 1,508 non-harmful entries written in Japanese, collected from unofficial school websites and fora. The original data was provided by the Human Rights Research Institute Against All Forms for Discrimination and Racism in Mie Prefecture, Japan. The entries were collected and labeled by Internet Patrol members (expert annotators) with the help of the government supplied manual [14]. The instructions given by the manual are briefly described below.

The definition given by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan suggests that cyberbullying occurs when a person is directly offended on the Internet. This includes publication of the person's identity, personal information and other aspects of privacy. Thus, as the first distinguishable features for cyberbullying, MEXT identifies private names (also initials and nicknames), names of organisations and affiliations and private information (address, phone numbers, personal information disclosure, etc.)

In addition, cyberbullying literature reveals vulgarities as one of the most distin-

[2] `https://www.kaggle.com/blackmoon/russian-language-toxic-comments`

guishing characteristics of cyberbullying [1, 105]. Also according to MEXT, vulgar language and cyberbullying can be distinguished from each other as cyberbullying conveys offenses against specific individuals. In the prepared dataset, all entries containing at least one of the above characteristics were listed as harmful.

The second Japanese dataset was collected from Twitter by Arata [127]. The dataset consists of random tweets that were collected during a one-week period in July of 2019. The collected information included the ID, the date and time of posting, the username, the text body, and the URL of the tweet. In addition, tweets written in other languages than Japanese, tweets submitted by bots and tweets consisting of less than five characters were filtered out from the collection.

The guidelines for the annotations were based on the information provided by Safer Internet Association [3] [4] and also on the research by Takenaka et al. [128]. The tweets were organized into seven different categories based on the above sources. The annotation work was carried out in two stages, primary annotation and secondary annotation. This was done in order to reduce the burden on each annotator and to not allow the work of each annotator influence one another. First, the annotators determined whether a tweet will fall into either harmful or not. After that, one of seven specific categories (illegal acts, prostitution, suicide, abuse/slander/bullying, obscenity, cruelty, other) was assigned to the tweet in the second annotation. In total, 30,425 tweets were annotated by six annotators who attended an Internet patrol course organized by the Hokkaido Police. For a single tweet, the primary annotation was done by a single annotator and the secondary annotation was carried out by two annotators. After this, all of the annotations were validated by at least three people.

The number of tweets annotated as harmful, defined in the guidelines mentioned above, was less than 2%. In addition, when looking at each category, about 75% of tweets were given the category of “abuse/slander/bullying". Furthermore, when validating the contents of tweets in this category, 202 tweets were additionally annotated as “prostitution", out of which almost half were duplicated. The 115 tweets that were not annotated in either of these categories were mostly related to “obscenity". In this study, a balanced random subset, consisting of 552 harmful

[3] https://www.saferinternet.or.jp/wordpress/wp-content/uploads/bullying_guideline_v3.pdf

[4] https://www.safe-line.jp/wp-content/uploads/safeline_guidelines.pdf

and 546 non-harmful entries, of the 30,425 tweets was used.

4.1.7 Korean Dataset

The Kaggle Korean Hate Speech Dataset [5] is a collection of Korean hate speech text data. Composed of hateful and discriminatory comments scraped from Korean alt-right website (Daily Best). The dataset was published on Kaggle in 2020. It consists of almost 190,000 comments, where 90,000 texts were labeled as hate speech, and 100,000 were labeled as normal. The average length of comments is around 50 characters. The annotators of this dataset are unknown so I am unable to say anything regarding its quality.

4.2 Methods

4.2.1 Models

For the experiments I used the following models. I assumed that these multilingual transformer models would be able to generalize sufficiently well in a zero-shot cross-lingual setting [129]. This means the fine-tuning is done without using any data from the target language. Instead, the fine-tuning is performed only with data from one of the other proposed languages.

Multilingual BERT (mBERT) [34] is the multilingual version of BERT, which stands for Bidirectional Encoder Representations from Transformers. It is based on the transformer [130], an attention mechanism that learns contextual relations between words (or sub-words) in text. One of the features transformer models introduced is the capability to read text input in both directions at once, instead of being able to only read sequentially from left-to-right or right-to-left. Taking advantage of this bidirectional capability, BERT is pre-trained on two NLP tasks, Masked Language Modeling and Next Sentence Prediction. The objective of Masked Language Modeling is to mask a word in a sentence and have the algorithm predict based on the word's context what word has been hidden. In Next Sentence Prediction, the objective for the algorithm is to predict whether two of

[5] `https://www.kaggle.com/captainnemo9292/korean-hate-speech-dataset`

any given sentences have a connection, either logical or sequential, or whether their relationship is random.

Even though mBERT has not been trained on any cross-lingual data, it has showed cross-lingual capabilities and had good results in many cross-lingual tasks [131]. This also includes various zero-shot transfer tasks. Multilingual BERT has even been shown to outperform the usage of various cross-lingual embeddings [132]. It is hypothesized that this generalization ability comes from having word pieces used in all languages (numbers, URLs, etc) which have to be mapped to a shared space, which in turn forces the co-occurring pieces to also be mapped to a shared space, thus spreading the effect to other word pieces, until different languages are close to a shared space [133].

XLM-RoBERTa (XLM-R) [134] is a fully cross-lingual transformer model, also trained on the Masked Language Model objective. XLM-R is trained on around a total of 2.5tb of CommonCrawl data in one hundred different languages. The model's training routine is the same as the monolingual RoBERTa model[121], meaning, that the sole training objective is the Masked Language Model. This means that the model is not trained on the Next Sentence Prediction task like in BERT or using the parallel Translation Language Model objective of XLM.

XLM-R has been shown to outperform both mBERT and XLM on a variety of cross-lingual benchmarks, which include zero-shot transfer tasks [135]. It also performs particularly well on low-resource languages. Interestingly, XLM-R is also very competitive when comparing to state-of-the-art monolingual models, which demonstrates that it is possible to create multilingual models without sacrificing per-language performance in a monolingual setting [134], most likely thanks to the sheer amount of data used in the pre-training.

4.2.2 Linguistic Similarity Metrics

To be able to calculate the correlation between cross-lingual zero-shot transfer performance and language similarity for the proposed tasks, I needed a way to quantify the aspects of all of the languages in the proposed set, specifically, a language similarity metric. I utilized three language similarity measures, eLinguistics [77], EzGlot [78] and the multidomain metric I quantified from the linguistic features presented in WALS [83].

eLinguistics [77] works by calculating a genetic proximity value for a pair of languages based on the use of phonetic consonants. The score is calculated by taking a predefined word set and comparing the consonants contained in these words. The method also takes into account the order of the consonants. This way, it is possible to get information regarding the closeness of the phonetics of the pair of languages set for comparison. The assessment of the relationship of the consonants is based on the research done by Brown et al [136].

Even though completely disregarding semantic, morphological, and syntactic similarity and being very simple in formulation, the similarity values produced by the method seemed to be in line with my intuition and the other metrics used in this research. However, as the distance between the two compared languages increased, the method seemed to become increasingly more prone to errors. This is due to the surging amount of accidental similarities in consonants. The similarity measure can be accessed from a web service [6]. The similarity values between the proposed languages are shown in Table 4.2.

Table 4.2: eLinguistics distance metric

	Danish	**English**	**German**	**Japanese**	**Korean**	**Polish**	**Russian**
Danish	0.0	20.6	38.2	95.2	97.2	68.2	66.2
English	20.6	0.0	30.8	88.3	90.0	66.9	60.3
German	38.2	30.8	0.0	87.4	95.5	68.1	64.5
Japanese	95.2	88.3	87.4	0.0	88.0	93.3	93.3
Korean	97.2	90.0	95.5	88.0	0.0	89.5	89.5
Polish	68.2	66.9	68.1	93.3	89.5	0.0	5.1
Russian	66.2	60.3	64.5	93.3	89.5	5.1	0.0

EzGlot [78] is based on the similarity of vocabularies, or lexical similarity, of the two compared languages. EzGlot's similarity metric is computed by taking the lexical similarity between the two compared languages, while in addition taking into account the number of words the pair of languages also have in common every other language. This makes it possible to compute a similarity measure for a pair of languages in relation to their closeness with every other language. Also, due to including the calculation of the number of words the languages share with all

[6] `http://www.elinguistics.net/Compare_Languages.aspx`

other languages, the similarity measure becomes asymmetric between every pair of languages. This also supports studies stating that mutual language intelligibility is being considered asymmetric as well [137, 32].

A pre-computed language similarity matrix and the formula for its computation can be found on the EzGlot similarity metric project's web page [7]. However, the usability of the metric is hindered by the high amount of missing values in the similarity matrix. For example taking a look at Japanese, which is one of the languages utilized in the experiments, over half of the values are missing for the proposed languages. Also, the authors of the similarity measure do not give away their data source. This means that I was unable to say anything regarding the quality of the computations. This also makes it more difficult to fill in the missing values to the similarity matrix. I extracted the similarity values from the EzGlot's similarity matrix for the proposed languages. These values are presented in Table 4.3.

Table 4.3: EzGlot similarity metric

	Danish	**English**	**German**	**Japanese**	**Korean**	**Polish**	**Russian**
Danish	100	9	17	N/A	9	13	N/A
English	6	100	28	7	26	19	14
German	6	15	100	N/A	5	8	4
Japanese	N/A	2	N/A	100	8	N/A	N/A
Korean	1	5	2	4	100	1	3
Polish	6	12	9	N/A	5	100	15
Russian	N/A	11	7	N/A	11	19	100

In addition, the plan was to take a look at the STL similarity measure [79], which is based on multiple linguistic features. The measure puts together three different aspects of language by using Semantic, Terminological (lexical) and Linguistic (syntactic) similarity to form a single metric. According to the authors, the STL metric outperformed many previous measures that were relying only on one of the previously mentioned feature types [80, 81]. However, in order to be able to use the metric, the vocabulary dataset must be structured in the form of an ontology, which restricts the metric's use. Due to this fact and because of the lack of available

[7] `https://www.ezglot.com/most-similar-languages.php`

languages for the used dataset, it was not possible to utilize the metric in this research.

Additionally, I considered using the lang2vec tool developed by Littell et al. [82], which is a database that represents languages as typological, phylogenetic, and geographical vectors. The method is based on multiple linguistic features, making it naturally more robust than EzGlot or eLinguistics similarity metrics, which only rely on a single linguistic feature each. However, due to being based on multiple sources, the heterogeneous nature of the method brings up many questions. For example, I do not know how features are selected from different sources and how they are weighted. Also, using geographical information as one of the vectors seems questionable as it was shown to be unreliable when predicting similarity of languages [87]. Instead, I decided to concentrate on quantifying a novel similarity metric from the World Atlas of Language Structures (WALS) which contains a variety of linguistic features from multiple domains such as phonological, grammatical and lexical.

4.2.3 The World Atlas of Language Structures

The World Atlas of Language Structures (WALS) [83] is a massive language database that records phonological, word semantic and grammatical information for a total of 2,662 languages from over 200 different language families. Theere are 192 different linguistic features in the database currently (May 2022). However, many of the linguistic features are missing for of the available languages. For example, one of the most extensively documented language, English, has about 150 features documented in the database. This amount rapidly decreases for languages studied less. Taking Danish as an example, it only 58 features documented [8]. Considering every language and all of the features, this adds up to over 58,000 data points in total in the WALS database. This means the whole database is only approximately 12% populated, meaning a vast majority of the information is missing. Also many major and widely studied languages are missing many features. For example, 25% of all of the features are missing for English. These missing values and the sparsity

[8]Some even less studied languages have an even smaller number of features documented, e.g. Chuj language, spoken in Guatemala, has only 29, while the Indonesian Kutai language has only a single feature documented.

of the data is the main point of concern when quantifying the WALS database into a linguistic similarity metric as using lesser known and not so widely studied languages means having less common features among them.

Another one of the goals of this study was to create a linguistic similarity metric, that would take multiple aspects of a language into account instead of only using a single feature, from the WALS database. To accomplish this I downloaded a snapshot and scanned through the database. Then selected all of the features that would have a defined value in all of the proposed languages (English, German, Danish, Polish, Russian, Japanese and Korean). This resulted in a total of 42 common features among the languages. Next I looked at the possible values each feature can take and converted the values for each feature to a numeric scale from zero to one in the relative order. To do this, I assumed that the order in which the possible values are presented in the WALS database roughly represents the order of their similarity. After converting the values, I used them to compare each language pair and calculate an average of the euclidean distances between all of the features for all of the possible combinations of languages, resulting in a symmetric distance metric. The finished distance matrix is shown in Table 4.4.

Table 4.4: WALS distance metric

	Danish	**English**	**German**	**Japanese**	**Korean**	**Polish**	**Russian**
Danish	0.000	0.098	0.086	0.252	0.212	0.154	0.152
English	0.098	0.000	0.149	0.306	0.240	0.155	0.152
German	0.086	0.149	0.000	0.332	0.292	0.181	0.179
Japanese	0.252	0.306	0.332	0.000	0.118	0.295	0.277
Korean	0.212	0.240	0.292	0.118	0.000	0.232	0.229
Polish	0.154	0.155	0.181	0.295	0.232	0.000	0.077
Russian	0.152	0.152	0.179	0.277	0.229	0.077	0.000

4.3 Experiments

4.3.1 Setup

I fine-tuned both of the models (mBERT, XLM-R) with all of the proposed languages (English, German, Danish, Polish, Russian, Japanese and Korean),

producing a total of 14 models. The fine-tuned models were then evaluated with test datasets from each of the proposed languages in order to calculate the cross-lingual transfer performances. The models were evaluated using a macro F1-score. After evaluating the models, I studied the correlation between classifier performance and language similarity using the previously introduced linguistic similarity metrics. Specifically, I calculated both Pearson's and Spearman's correlation coefficients between the models and the linguistic similarity metrics. The training of the classifiers was done using PyTorch and the Transformers library on an Nvidia GTX 1080Ti.

4.3.2 Classification Results

I fine-tuned the multilingual transformer models with the offensive language datasets described earlier. Each model was fine-tuned only on a single language before evaluation. The classifier evaluation results were shown in Tables 4.5 and 4.6.

Table 4.5: Classification scores (F-score) for Multilingual BERT

		Target						
		Danish	**English**	**German**	**Japanese**	**Korean**	**Polish**	**Russian**
Source	**Danish**	0.75	0.54	0.50	0.37	0.40	0.51	0.55
	English	0.53	0.77	0.41	0.33	0.34	0.44	0.41
	German	0.57	0.56	0.70	0.48	0.45	0.63	0.72
	Japanese	0.50	0.53	0.46	0.88	0.49	0.49	0.51
	Korean	0.43	0.44	0.33	0.45	0.95	0.38	0.50
	Polish	0.51	0.48	0.41	0.33	0.34	0.83	0.58
	Russian	0.50	0.47	0.61	0.64	0.61	0.60	0.90

From the results, it is clear that XLM-R outperformed mBERT, as its scores are higher across the board. Also, the scores are obviously highest when using the same language as source and target. The second highest scores are usually by the languages in the same language families (English, German, Danish - Germanic; Polish, Russian - Slavic; Japanese, Korean - Koreano-Japonic).

As can be seen from Tables 4.5 and 4.6, English was generally the worst language to use as the source, having low scores with all languages but itself. German and

Table 4.6: Classification scores (F-score) for XLM-RoBERTa

		Target						
		Danish	**English**	**German**	**Japanese**	**Korean**	**Polish**	**Russian**
Source	**Danish**	0.75	0.67	0.49	0.39	0.39	0.56	0.57
	English	0.58	0.81	0.43	0.46	0.41	0.47	0.45
	German	0.66	0.66	0.73	0.55	0.52	0.64	0.71
	Japanese	0.56	0.57	0.47	0.90	0.61	0.55	0.65
	Korean	0.43	0.49	0.34	0.52	0.95	0.39	0.48
	Polish	0.63	0.60	0.50	0.46	0.50	0.84	0.78
	Russian	0.57	0.53	0.66	0.72	0.76	0.65	0.91

Danish worked best as the source language for English. In general, German worked well as a source language but was hard to generalize on by other languages. For example, both English and Danish worked better as a source for Polish and Russian than German. In addition, German worked especially well as a source language for the Slavic languages (Polish and Russian). Also, Russian had the highest zero-shot performance as the source language for German. Also, German was the best source for Danish. Polish received a good score as the source language for Russian. However, Russian did not do so well as the source for Polish, being equalled by German. Interestingly, Russian had good scores as the source language for Japanese and Korean, a feature that no other language had, outperforming both Japanese and Korean when used as source languages for one another. Specifically, Russian to Japanese yielded a score of 0.64 while Korean to Japanese was only 0.45, and the score for Russian to Korean was 0.61 while Japanese to Korean was 0.49 with XLM-R. Generally, Japanese and Korean were the hardest target languages and interestingly, did not score well as a language pair despite being classified in the same language family.

4.3.3 Correlation with Linguistic Similarity

I calculated Pearson's and Spearman's correlation coefficients (ρ-value) between the classification results of the two classifiers and each of the three proposed linguistic similarity metrics (EzGlot, eLinguistics, WALS). As the similarity matrix for EzGlot was not fully populated, I needed to ignore the scores for the missing language pairs during the calculation of the correlations for this particular similarity

metric. The results can be seen in Tables 4.7 and 4.8 for Pearson's and Spearman's correlation coefficients respectively.

Table 4.7: Pearson's correlation coefficient for classifier scores and linguistic similarity metrics

	XLM-R		**mBERT**	
	ρ	**p-value**	ρ	**p-value**
WALS	-0.674	0.001	-0.713	0.001
EzGlot	0.720	0.001	0.801	0.001
eLinguistics	-0.713	0.001	-0.736	0.001

Table 4.8: Spearman's correlation coefficient for classifier scores and linguistic similarity metrics

	XLM-R		**mBERT**	
	ρ	**p-value**	ρ	**p-value**
WALS	-0.599	0.001	-0.615	0.001
EzGlot	0.506	0.001	0.494	0.001
eLinguistics	-0.654	0.001	-0.666	0.001

As can be seen from the results, Pearson's correlation is strong with all of the proposed similarity metrics. Also, the p-value is less than 0.05 in all of the cases, showing that the results are statistically significant. EzGlot's similarity metric has the strongest correlation with $\rho = 0.720$ for XLM-R and $\rho = 0.801$ for mBERT. This is followed by eLinguistics and WALS with the absolute correlation in the range of 0.67 to 0.73. Spearman's correlation is slightly lower, being in the moderate-strong range with all of the metrics, with the p-value also being less than 0.05. The strongest correlation is by eLinguistics with an absolute correlation of $\rho = 0.654$ for XLM-R and $\rho = 0.666$ for mBERT. This is followed by eLinguistics and WALS with the absolute correlation in the range of 0.49 to 0.62. Also, the correlations were generally slightly stronger with mBERT than with XLM-R.

However, after removing the same source-target language pairs and leaving only the zero-shot classification results, the correlations changed drastically. This is shown on Tables 4.9 and 4.10 for Pearson's and Spearman's correlation coefficients,

respectively. There were two changes. First, EzGlot's similarity metric plummeted down from having the strongest correlation with Pearson's to showing no correlation at all, both correlation coefficients showing a value near zero, and losing statistical significance. Second, the correlations for eLinguistics and WALS also fell from strong to moderate, standing now in the range of 0.35 to 0.44 for Pearson's and 0.37 to 0.48 for Spearman's correlation coefficient. The p-values also increased slightly, but remained under 0.05, keeping statistical significance. Also, the correlation of eLinguistics stayed stronger than that of WALS despite the other changes.

Table 4.9: Pearson's correlation coefficient after removing the same source-target language pairs

	XLM-R		**mBERT**	
	ρ	**p-value**	ρ	**p-value**
WALS	-0.359	0.020	-0.368	0.016
EzGlot	0.011	0.953	-0.040	0.829
eLinguistics	-0.438	0.004	-0.421	0.006

Table 4.10: Spearman's correlation coefficient after removing the same source-target language pairs

	XLM-R		**mBERT**	
	ρ	**p-value**	ρ	**p-value**
WALS	-0.377	0.014	-0.395	0.010
EzGlot	0.129	0.480	0.100	0.584
eLinguistics	-0.465	0.002	-0.475	0.001

4.4 Discussion

4.4.1 Transfer Language Performance

The fact that XLM-R outperformed mBERT matches expectations, as it also did so on a variety of benchmark tasks [90, 91]. The reasons are most likely that XLM-R is a true cross-lingual model and has a vastly larger vocabulary size

than mBERT. The highest cross-lingual transfer scores (Tables 4.5 and 4.6) were usually by languages from the same language families as the source language. This matches with the typical intuitive selection process when selecting the transfer source language. However, this was not always the case and one relying purely on selecting the languages from the same language family will lead to diminished transfer performance in some cases. A good example of this is when the target language is German. One could expect the best transfer languages being Danish and English, but actually both Polish and Russian had a higher transfer performance despite of being from the Slavic language family, not Germanic. This could be due to the differences in grammatical complexity. Both Danish and English have relatively simple grammar compared to German, which could leave them unable to generalize on the more sophisticated German language. On the other hand, the grammar of Polish and Russian is even more complex, which could negate this issue, allowing them to generalize better. Also, German, Polish and Russian all have synthetic morphology, which could play a role in their ability to generalize well on each other. Furthermore, the historical mutual influence between Germans, Poles and Russians could be a factor here. Looking at the scores, it can be noted that German is a good source for both Germanic and Slavic languages.

English on the other hand was one of the worst, if not the worst language to use as the transfer source overall. It had a poor performance even in its own language family, probably due to its simplicity when compared to both Danish and German. Also, English is heavily influenced by French, further distancing it from the other Germanic languages. Furthermore, the differences in morphology could be a factor. The analytic nature of English could be a reason why it cannot generalize on fusional languages like German. Danish, which is also an analytic language had a better generalization for German probably due to its otherwise closer ties to the German language (eg., mutual influence). These results show that other languages should be considered over English as the cross-lingual transfer source if available.

Interestingly, Russian achieved a high score as the transfer language source for both Korean and Japanese. Which is different from any of the other languages included in this study. The reason could be in the shared morphological features, specifically the fact that all of the three languages are agglutinative. Furthermore,

all of the three languages contain distinct registers, specifically for expressing different politeness levels. However, as this could be heavily related to the topic of offensive language identification or to the properties of these specific datasets and might not be applicable to other fields, this needs to be confirmed in the future on other datasets and, preferably, different tasks as well.

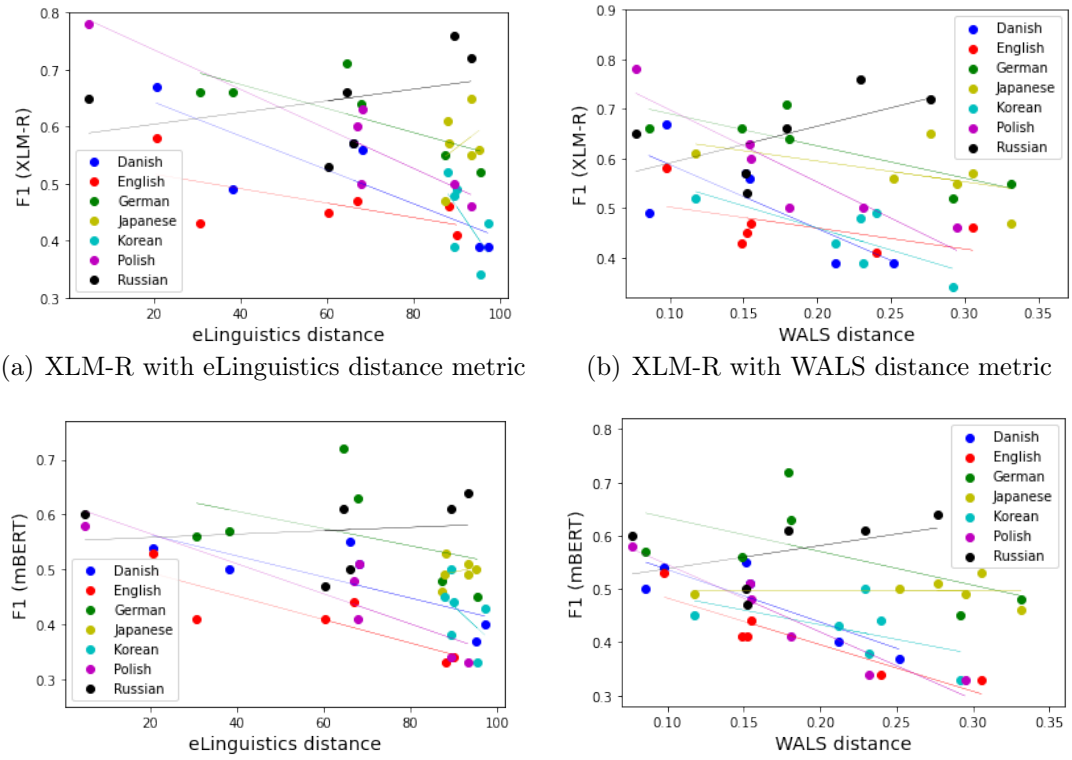
Also, Korean and Japanese did not work as well with each other, contrary to how I was expecting, and were clearly outperformed by Russian, even though they are more similar with one another than with any of the other languages used in this study. This as well could be related to the properties of the datasets or to the topic of offensive language identification itself and will have to be verified in the future.

Furthermore, when looking at the source languages individually, one could see a trend of the transfer performance being better when transferring to more similar languages in all cases except Russian, as it tended to have an exceptionally high transfer performance to both Japanese and Korean when compared to other languages. For all of the other languages, the performance clearly decreases as the distance between the languages increases. This can be seen from Figure 4.1.

The sizes of other datasets vary from around 3,000 samples to almost 200,000 samples. Also, the percentage of harmful samples in each dataset varies greatly. The English dataset has only 7% of harmful samples while Japanese and Korean have around 50%. In order to determine, whether the dataset size and balance had an effect on the results or not, I decided to further analyze the effectiveness of each dataset as the transfer source. In order to measure the effect, I calculated the correlation between the average transfer scores of each dataset and the dataset size/balance by taking a base-10 logarithm of the dataset size, multiplied by the proportion of harmful samples in the dataset. I had to take the logarithm of the sample size, because otherwise its weight would become too large compared to the harmful ratio, which is bound between 0 and 1. The results are shown in Table 4.11.

As can be seen from the results, mBERT and XLM-R show no correlation between the average performance of the classifier as the source language and the size/balance of the dataset. With this I can more safely say that the results should not be biased by the differences in the size and balance of the datasets.

z



(a) XLM-R with eLinguistics distance metric

(b) XLM-R with WALS distance metric

(c) mBERT with eLinguistics distance metric

(d) mBERT with WALS distance metric

Figure 4.1: Performance trends for source languages for cross-lingual transfer

Table 4.11: Upper: average F1 and dataset size and balance statistics, lower: Pearson's correlation coefficient for average F1 and size/balance

	mBERT	**XLM-R**	**Samples (nS)**	**Harmful ratio (Hr)**	**log(nS)*Hr**
Danish	0.48	0.51	3,289	0.13	0.45
English	0.41	0.47	12,772	0.07	0.29
German	0.57	0.62	8,407	0.34	1.32
Japanese	0.5	0.57	6,434	0.50	1.91
Korean	0.42	0.44	189,995	0.47	2.50
Polish	0.44	0.58	34,953	0.21	0.96
Russian	0.57	0.65	14,412	0.33	1.39
Pearson ρ	0.115	0.111			
p-value	0.977	0.812			

4.4.2 Analysis of Specific Examples

In order to clarify the characteristics of the classification models, I chose a number of example sentences from the English cyberbullying test dataset based on the prediction results and confidence (strength of prediction by the model). I inspected the results and tried to reason why the model made the decisions, concentrating on potential points of failure. The texts were chosen by taking three properties into account, confidence (low/high), label (0/1) and prediction result (incorrect/correct), resulting in eight examples. The results can be found in Table 4.12.

Table 4.12: Example sentences, predictions and confidence values (XLM-R)

Text	Label	Pred (en)	Conf (en)	Pred (da)	Conf (da)	Pred (de)	Conf (de)	Pred (pl)	Conf (pl)	Pred (ru)	Conf (ru)	Pred (ja)	Conf (ja)	Pred (ko)	Conf (ko)
I GUNA PEEEEEEEE MY PANTS CUZ OV U go wee then sori	0	1	0.52	0	0.71	0	0.58	0	0.91	1	0.75	0	0.84	0	0.64
Are you a nun Ummm Nopee... Dont Thinkk Soo.. :)	0	1	0.83	0	0.75	0	0.83	0	0.94	1	0.62	0	0.83	0	0.76
brittany is from MAYODAN abbie aint from ellisboro either.dumbass....and the lst 2 aintchur girls they dont evn lik u. I was naming people who live out in that area ha yeahh okayy. I really don't care what some bitch thinks about me (:	1	0	0.55	1	0.70	1	0.69	0	0.92	1	0.77	1	0.67	0	0.51
what do you think about men who like to be dominated and ridiculed in bed? LMAO suckers ! !	1	0	0.93	1	0.57	1	0.65	1	0.69	1	0.79	1	0.76	0	0.89
—fuck me— hah... nah stranger;)	1	1	0.64	1	0.63	0	0.81	0	0.95	1	0.57	0	0.77	0	0.83
you can suck the dick bitch dont try me little girl dany shut the fuck upppp BITCH...r r	1	1	0.87	1	0.77	1	0.62	0	0.82	1	0.79	1	0.85	1	0.81
i love how the people who talk shit are all anonymous. they're scared of youu (:r lolll. lofl i fukin noee rytee;]	0	0	0.50	1	0.69	0	0.53	0	0.91	1	0.74	0	0.57	1	0.53
Shouldn't the opposite of shut up be shut down? i guesss so. ha	0	0	0.94	0	0.88	0	0.83	0	0.95	0	0.76	0	0.81	0	0.78

Looking at the results, the low confidence texts seem to have two factors in common. First, they contain many slang words, spoken contractions and typos. These words are most likely not contained in the pretrained model's vocabulary and thus make the texts harder to classify. Second, the annotations of these texts seem more ambiguous and could be interpreted in different ways. For example,

looking at the third example it is not impossible to see why it has been labeled as it is (revealing someone's location by city and using vulgar language), but one could also argue that these alone are not enough to label the text as cyberbullying. Also, the expression used in the fifth text could mean either sexual demand or be a sarcastic comment about a frustrating situation. Although here the context implies it is sexual.

The mistakes the model made with high confidence are more difficult to reason. For example, the second example does not imply anything cyberbullying related but the model still predicted it as so. In the future, I plan to investigate the attentions put on each tokens to be able to produce a working theory of why the model makes such decisions. Similar thing applies to the fourth sentence, although the label and the prediction are the opposite. The sixth and eight examples were correctly classified with high confidence and they clearly represent their assigned labels.

Looking at the predictions and confidences of the models fine-tuned in other languages than English, the situation is quite different. It looks like the other models performed better on these texts than the English model. For example, the Danish, German and Japanese models classified seven out of these eight texts correctly. Also, the confidences seem to vary a lot, with both close and distant languages. In the future, I plan to take a deeper look into the cross-lingual classification results and investigate, what could be the possible causes for the model's behaviour.

4.4.3 Analysis of Linguistic Similarity Metrics

The correlation of EzGlot's similarity metric was higher than those of eLinguistics' or WALS' as can be noted from Tables 4.7 and 4.8. Being based on lexical similarity, it would suggest that the used multilingual models heavily rely on lexical information. However, when considering only the zero-shot classification results (Tables 4.9 and 4.10), EzGlot's similarity metric changed to showing no correlation with the transfer performance at all. This shows that they do not rely only on lexical features and that other linguistic features need to be considered when choosing the source language for cross-lingual transfer.

To my surprise, the correlation of eLinguistics' metric was higher than the correlation of the multidomain metric I quantified from WALS despite of being calculated only by comparing a predetermined set of phonetic consonants. Possibly,

the choice of including only the features common among all of the proposed languages could have caused too many irrelevant features to be included. This might have resulted in bias in the metric calculation. In the future, I will aim for a better quantification of the WALS database in order to develop an even more effective and comprehensive similarity metric, also by incorporating the other two metrics.

Table 4.13: Target languages, best sources and their similarity ranks for eLinguistics and WALS metrics

Target	**Best source**	**eLinguistics**	**WALS**
Danish	German	2	1
English	Danish	1	1
German	Russian	3	3
Japanese	Russian	4	4
Korean	Russian	2	3
Polish	Russian	1	1
Russian	Polish	1	1

However, even though having the highest correlation, the eLinguistics metric has its weaknesses due to being based on only one aspect of language. Looking at Table 4.2 one can see that eLinguistics shows Japanese being very distant from Korean, being at the same level as Polish and Russian, which is in fact not true due to similarities in vocabulary and grammar between Japanese and Korean. The WALS metric on the other hand is obviously more robust to errors like this as can be seen from Table 4.4. This is most likely thanks to it being based on multiple linguistic features instead of only one as is the case with eLinguistics metric. Table 4.13 shows that on average the metrics look equally good if they were used for transfer language selection. Only difference being that when using WALS, the best transfer option was chosen more often.

The fact that the similarity metrics of eLinguistics and WALS correlated with transfer language efficacy means that they can be used for the selection process. So, instead of making a decision based on intuition or simply choosing any language from the same language family, one can check the similarity of the target language with high-resource languages that have proper data available and make a more informed and effective decision, at least for offensive language identification. This

allows for more efficient model development.

4.4.4 Ethical Considerations

Being able to choose an optimal transfer language can greatly aid in the task of detecting harmful language like hate speech and cyberbullying, especially when dealing with low-resource languages. The fact that there are thousands of languages used every day in online communication and social media, of which only a small fraction have proper data to train the detection models on, shows the amount of potential this method has. This will make it possible to detect offensive content as early and effectively as possible to prevent its serious consequences and control its spread. The method ultimately aids in reducing the damages abusive and harmful content causes to the society. Also, it can reduce the human effort required to keep offensive content like cyberbullying and hate speech at bay, release the society's resources for development in other fields and ease the burden on those who have to deal with this serious problem in any way.

To take a look at the ethicality of the used transformer models, I need to inspect the data used for their pretraining. The corpora used for pretraining are from Wikipedia (mBERT) and CommonCrawl (XLM-R) [134]. These differ greatly in domain as Wikipedia consists only of well structured documents written in formal language, whereas CommonCrawl, being basically a snapshot of the Web, might contain almost anything from structured text (Wikipedia) to more natural texts like blogs. This means that while Wikipedia is already mostly internally detoxified due to its ethical guidelines and moderation, the CommonCrawl data most likely contains also unethical matter not only because of the inclusion of more natural texts like blogs or product reviews, but also as the result of media bias induced by the addition of news data. Paradoxically, the fact that XLM-R is also pretrained on possibly toxic content could be a contributing factor to its higher performance in the cyberbullying detection task. In order to investigate the effect of including possibly unethical material in model pretraining, I calculated the correlation between the proportion of possibly unethical text (non-Wikipedia) in the corpus and classifier performance.

As the exact amounts of Wikipedia data used in the pretraining were not available, I ranked the proposed languages based on the approximate proportions

[134] of unethical text and calculated Spearman's rank correlation coefficient between the approximate proportions and the classifier scores[9]. This resulted in Spearman $\rho = 0.036$, meaning that I could not find a correlation between using possibly unethical data in pretraining and the performance of a fine-tuned model. However, due to the limited scope of this research, further study is required to investigate the effect of including unethical text in the pretraining process.

In addition to containing texts from different domains, the corpora also holds different amount of data for different languages. This could cause initial bias in the language coverage, which could also have an impact on the performance. Looking at the amount of data used from the proposed languages, the pretraining corpora sizes with XLM-R vary from 300GB (English) to 45GB (Polish). In order to determine whether the results are biased by the amount of pretraining data, I calculated the correlation between the pretraining corpus size and classifier performance. The results were shown in Table 4.14.

Table 4.14: Upper: Pretraining corpus size and average classifier performance (F1), lower: Pearson's and Spearman's correlation coefficient for pretraining corpus size and average F1

Language	**Size (GB)**	**F1 (XLM-R)**
English	300.8	0.47
Russian	278.0	0.65
Japanese	69.3	0.57
German	66.6	0.62
Korean	54.2	0.44
Danish	45.8	0.51
Polish	44.6	0.58
Pearson	ρ: 0.085	p-value: 0.86
Spearman	ρ: 0.071	p-value: 0.88

In the case of the proposed languages, I was not able to notice that the pretraining corpus size influenced the results. However, further research is required to investigate the possible bias in pretrained multilingual models considering the used pretraining

[9]Here, I used only Spearman's rank correlation coefficient without Pearson's correlation coefficient, since, although it was possible to deduce the relative approximate proportions, the exact numbers were not mentioned in the literature.

corpora sizes for each language.

4.4.5 Limitations

Naturally, the pretraining data size should have an effect to the classifier's performance even though I didn't find any correlation. One of the factors in the overall high performance of the Russian model could be the high pretraining data size. However, the English model performed poorly despite of being pretrained on the largest corpus. Unfortunately it is impossible to account for this unless I pretrain the models ourselves. The point of these experiments was to concentrate on optimizing the usage of existing resources, but I consider pretraining the models with similar corpora for the next steps of the research in order to have a more uniform setup and to reduce the amount of confounding factors.

Another factor is the differences between the fine-tuning datasets. The dataset size and the ratio of positive and negative class being one that could bias the results. However, I couldn't find a correlation between these factors and the classifier performance. Another difference comes with the dataset domains. Cyberbullying, hate speech and toxic language have their innate differences and could impact the transfer performance in a cross-domain case [138]. Unfortunately fixing these issues would require the collection and annotation of all of the datasets as quality datasets are already scarce when considering many of the used languages. Also, the main goal of the paper was to aid in the creation of offensive language detection models for especially low-resource languages using what is currently available. In the future, when one becomes available, the plan is to repeat the experiments on a fully cross-lingual offensive language dataset created as uniformally as possible apart from language.

4.4.6 Future Research

In the future, I am planning to re-quantify the WALS database in order to develop an even more effective and comprehensive similarity metric. In the current implementation, many features were cut out due to the data being too sparse. One option here would be to calculate the features separately for each language pair. This would allow to capture more features per language, but would lead to

inconsistencies when comparing languages as a different amount of features would be used. Another solution would be to determine which features have the strongest correlation with the cross-lingual transfer performance and then create the metric based on those features. Also, it would be a great aid if the WALS project received more attention and the feature matrix became more populated.

I am hypothesizing that this method could be useful as a general method also for other Natural Language Processing tasks outside of offensive language identification. This would help the model developer to make a more effective and justifiable decision instead of relying on intuition or simply choosing a language from the same family. I need to confirm the effectiveness of the selection method also for other tasks like sentiment analysis [139, 140], dependency parsing [141, 132, 142], named entity recognition [132, 143]. This will be explored in the next Chapter.

I plan to implement the method in the development of a multilingual cyberbullying detection application. With a proper transfer language selection procedure, it is possible to deal with some of the difficulties encountered earlier with languages lacking data. Furthermore, the linguistic features described in WALS could help find new insights about the features of offensive content in order to further practical research on cyberbullying and hate speech and ways of their mitigation.

Chapter 5

Generalization of Transfer Language Selection Method

In this Chapter, I propose to investigate the possibility that different linguistic similarity metrics could be utilized when trying to find possible source language candidates for cross-lingual transfer also for other tasks than abusive language detection. I hypothesize that linguistic similarity correlates with cross-lingual transfer efficacy, meaning that by using more similar languages one would achieve higher model performance.

In this research I concentrated on three different Natural Language Processing tasks, namely, sentiment analysis, named entity recognition and dependency parsing. I used datasets from eight different languages, namely English, German, Danish, Polish, Croatian, Russian, Japanese and Korean. The languages were chosen as they have relatively high quality datasets available. Also, the languages represent different language families (English, German, Danish - Germanic; Polish, Russian, Croatian - Slavic; Japanese, Korean - Koreano-Japonic language family). This also gives the opportunity to further study the efficacy of cross-lingual transfer learning between and within language family groups.

In practice, similarly as in the previous Chapter, I fine-tuned mBERT and XLM-R, this time on three different Natural Language Processing tasks (sentiment analysis, named entity recognition and dependency parsing) using eight different languages (English, German, Danish, Polish, Russian, Croatian Japanese, Korean) and then perform zero-shot prediction on the rest of the languages of the proposed

set. I calculated the linguistic similarity between all of the proposed languages using three different linguistic similarity metrics, EzGlot, eLinguistics and an improved version of the quantified model based on the World Atlas of Language Structures. I then calculated the correlation between the zero-shot cross-lingual transfer performance and linguistic similarity to show the effectiveness of the method.

5.1 Tasks

This research concentrate on three different NLP tasks. Sentiment analysis as a document classification task. Named entity recognition as a token classification task. And lastly, dependency parsing for understanding the importance of syntax and grammar in cross-lingual transfer.

I hypothesize that the zero-shot cross-lingual transfer performance correlates with the linguistic similarity of the source and target languages. In order to confirm the hypothesis, I used datasets from eight different languages, namely English, German, Danish, Polish, Russian, Croatian, Japanese and Korean for all of the tasks. I chose these languages as they had high quality datasets compared to other options and because the languages represent three different language families (English, German, Danish - Germanic; Polish, Russian, Croatian - Slavic; Japanese, Korean - Koreano-Japonic language family). This also gives the opportunity to study the efficacy of cross-lingual transfer learning between and within different language family groups.

5.1.1 Sentiment Analysis

In the field of NLP, sentiment analysis is one of the most active research areas [144]. The recent research in sentiment analysis, as with many other NLP tasks, has mainly focused on using deep neural networks and pretrained language models [145, 146, 147, 148, 149]. The popularization of multilingual transformer models has made it possible to utilize cross-lingual transfer in order to train models for low-resource languages.

Rasooli et al. [150] used a set of 16 languages from different language families,

namely Indo-European, Turkic, Afro-Asiatic, Uralic, and Sino-Tibetan, to learn a sentiment analysis model. Their experiments showed that for most target languages the best result can be obtained by leveraging from multiple source languages at the same time. Also, datasets of a similar genre and domain tended to yield higher results when compared to out-of-domain and dissimilar genres.

Pelicon et al. [151] used zero-shot cross-lingual transfer to classify Croatian news articles with an mBERT model fine-tuned using Slovene data with good results. In addition, Kumar et al. [152] used XLM-R and performed cross-lingual transfer from English to Hindi. Their model compared favorably to the used benchmarks and gives an effective solution to the analysis of sentiments in a resource-poor scenario.

The majority of the sentiment analysis datasets used in this research consists of product reviews, as I attempted to keep the domain the same throughout the languages. However, for some languages, I was unable to find such data, most notably Croatian, which consists of news articles. I also had to adjust the labels of some of the datasets so that they would match among all of the languages. Training and evaluation splits were retained from original datasets if possible, otherwise datasets were split to 80% training and 20% evaluation.

For this research I used the Multilingual Amazon Reviews Corpus [153], which covers English, Japanese and German from the proposed languages. The dataset contains over 200,000 reviews for each language collected between 2015 and 2019. The reviews are labeled from one to five stars. However, as the other datasets used in this research used a two-point scale (positive, negative), I adjusted the labels accordingly (positive: 5 and 4 stars, negative: 2 and 1 stars).

For Danish, I used a dataset containing almost 45,000 reviews crawled from Trustpilot by Alessandro Gianfelici [1]. For Polish, I used the PolEmo 2.0 corpus [154]. This dataset contains over 8,000 reviews from the domains of medicine, hotels, products and school. For both of these datasets, I also had to adjust the labels of this dataset to a two-point scale similarly to the Amazon Reviews dataset.

The Russian dataset used in this research was a product review dataset by Smetatnin et al. [155]. The dataset consists of 90,000 automatically labeled reviews on the topic "Women's Clothes and Accessories", split evenly among three classes (positive, neutral, negative). The Croatian dataset is the same used by Pelicon et

[1] `https://github.com/AlessandroGianfelici/danish_reviews_dataset`

al. [151], containing around 2,000 news articles. The articles were collected from 24sata, one of the leading Croatian media companies. The annotations were done by 6 people using a five-level Likert scale. The annotations were later adjusted to a three-point scale by the authors. For the purpose of the experiments, in case of both datasets, I left out the neutral reviews in order to binarize the labels.

The Korean dataset used in this research was Naver sentiment movie corpus v1.0 [2]. The dataset consists of Naver Movie reviews, with 100,000 positive and negative samples. The reviews were originally rated from one to ten, but the creators binarized the dataset prior to publishing.

5.1.2 Named Entity Recognition

The research on Named Entity Recognition (NER) has also shifted towards using Deep Neural Networks and most recently, pretrained transformer models [156, 157, 158]. Cross-lingual transfer has also been applied to NER in multiple research. Fritzler et al. [159] used a metric-learning method to at the time outperform a state-of-the-art recurrent neural network method and showed to be capable in both few-shot and zero-shot settings. Moon. et al. [160] used multilingual BERT to fine-tune a NER model in multiple languages and showed it to be more effective than a model fine-tuned only on a single language. This demonstrates that the model can leverage knowledge from other languages in order to improve its performance on one.

Hvingelby et al. [161] presented a Danish NLP resource based on the Danish Universal Dependencies treebank and showed that transferring from other Germanic languages, especially from English and Norwegian, to Danish can yield good results when using mBERT. However, using other Germanic languages in addition to Danish did not give any better results compared to fine-tuning only with Danish in their case.

Entity projection [162, 163] has been used to generate pseudo-labeled datasets for low-resource NER datasets with the help of parallel corpora. However, it has been shown by Weber and Steedman [164] that entity projection can be outperformed by cross-lingual transfer and XLM-RoBERTa. The reason behind

[2] `https://github.com/e9t/nsmc`

this could be explained by the discovery by Lauscher et al. [101], who showed that transfer performance with English as the source correlates with the similarity of the languages when dealing with a NER task.

In this study, I used the WikiANN [165] multilingual NER dataset also used by XTREME benchmark [90] for all of the proposed languages. WikiANN consists of Wikipedia articles annotated with LOC (location), PER (person), and ORG (organisation) NER tags. I used the version by Rahimi et al. [166], which has a balanced train, development, and test splits and supports 176 of the 282 languages from the original WikiANN corpus.

5.1.3 Dependency Parsing

Cross-lingual transfer in dependency parsing (DEP) has been studied for some time before the advent of multilingual transformer models [167, 168, 169, 170]. These studies mainly used deep neural network-based methods on parallel corpora. The research by Duong et al. [95] discussed earlier in Section 2.3.2 was also conducted on a dependency parsing task. Instead of using parallel corpora, their research was built around syntactic cross-lingual word embeddings [171] trained over POS contexts to emphasize syntax.

Multilingual transformer models have also seen success in the dependency parsing task [132, 172, 173]. Most notably, in their study, Lauscher et al. [101] discovered that structural and syntactic similarities between languages are the most determining factor when it comes to the success of cross-lingual transfer for lower-level tasks like POS-tagging and DEP.

The dataset used for all of the proposed languages in this study was the Universal Dependencies v2 [174], a widely used resource in NLP as well as in linguistic research. The dataset was also used in the XTREME [90] benchmark and in the research by Lauscher et al. [101] described earlier. Universal Dependencies is a framework for a consistent annotation of grammar, including parts-of-speech, morphological features, and syntactic dependencies across a total of more than 100 languages.

5.2 Experiments

5.2.1 Setup

For the experiments I used the same models as in the previous Chapter (mBERT, XLM-R) in a zero-shot setting with the three proposed tasks. In order to measure the correlation between cross-lingual transfer performance and linguistic similarity, we used eLinguistics [77], EzGlot [78] and the metric quantified from WALS [83]. These are the same metrics I used in the previous Chapter. The similarity matrices are shown in Table 5.1 for eLinguistics, Table 5.2 for EzGlot and Table 5.3 for the metric quantified from WALS.

Table 5.1: eLinguistics metric between all applied languages

	Danish	**English**	**German**	**Croatian**	**Polish**	**Russian**	**Japanese**	**Korean**
Danish	0.00	20.60	38.20	66.20	68.20	66.20	95.20	97.20
English	20.60	0.00	30.80	60.30	66.90	60.30	88.30	90.00
German	38.20	30.80	0.00	64.50	68.10	64.50	87.40	95.50
Croatian	66.20	60.30	64.50	0.00	10.70	5.60	90.70	87.20
Polish	68.20	66.90	68.10	10.70	0.00	5.10	93.30	89.50
Russian	66.20	60.30	64.50	5.60	5.10	0.00	93.30	89.50
Japanese	95.20	88.30	87.40	90.70	93.30	93.30	0.00	88.00
Korean	97.20	90.00	95.50	87.20	89.50	89.50	88.00	0.00

Table 5.2: EzGlot metric between all of the proposed languages

	Danish	**English**	**German**	**Croatian**	**Polish**	**Russian**	**Japanese**	**Korean**
Danish	100	9	17	N/A	13	N/A	N/A	9
English	6	100	28	6	19	14	7	26
German	6	15	100	4	8	4	N/A	5
Croatian	N/A	4	5	100	14	9	N/A	5
Polish	6	12	9	14	100	15	N/A	5
Russian	N/A	11	7	11	19	100	N/A	11
Japanese	N/A	2	N/A	N/A	N/A	N/A	100	8
Korean	1	5	2	1	1	3	4	100

In previous Chapter, I quantified a novel linguistic similarity metric from the WALS database based on the features all of the proposed languages shared. One

of the problems of the metric was that as the amount of languages increased, the amount of features shared with them decreased due to missing values in the database. This time I attempted to counter this by selecting all of the features that would have a defined value for both languages in all possible language pairs instead of having to be shared between all of the languages. The language pairs were formed from the proposed languages (English, German, Danish, Polish, Russian, Croatian, Japanese and Korean). Otherwise, the process remained the same. In short, I converted the possible feature values into numeric and calculated an average euclidean distance between all language pairs.

Table 5.3: WALS metric between all of the proposed languages

	Danish	**English**	**German**	**Croatian**	**Polish**	**Russian**	**Japanese**	**Korean**
Danish	0.000	0.109	0.140	0.167	0.197	0.155	0.236	0.202
English	0.109	0.000	0.136	0.179	0.164	0.141	0.252	0.209
German	0.140	0.136	0.000	0.221	0.196	0.140	0.248	0.225
Croatian	0.167	0.179	0.221	0.000	0.160	0.080	0.272	0.229
Polish	0.197	0.164	0.196	0.160	0.000	0.097	0.249	0.210
Russian	0.155	0.141	0.140	0.080	0.097	0.000	0.225	0.196
Japanese	0.236	0.252	0.248	0.272	0.249	0.225	0.000	0.108
Korean	0.202	0.209	0.225	0.229	0.210	0.196	0.108	0.000

In practice, I fine-tuned both of the models (mBERT, XLM-R) with all of the proposed languages (English, German, Danish, Polish, Russian, Croatian, Japanese and Korean) for all of the tasks. This produced a total of 16 models for each task, a total 64 models. After fine-tuning, I evaluated the models with test datasets from all of the previously mentioned languages to compute the cross-lingual zero-shot transfer scores. I did not use a train-dev-test, but only train-test scenario for evaluation, because the test dataset has nothing to do with the training dataset in a zero-shot task. I also do not aim at optimizing for each dataset, or creating a product, but rather study general properties. I evaluated the models with a macro F1-score for sentiment analysis and NER, and Label Attachment Score (LAS) for the dependency parsing task.

After finishing the evaluations for all of the fine-tuned models, similarly as in the previous Chapter, I took a look at the correlation between the zero-shot cross-lingual transfer scores and linguistic similarity. This was done by using the

three previously introduced linguistic similarity metrics (eLinguistics, EzGlot and WALS). I computed Pearson's and Spearman's correlations between the models' cross-lingual zero-shot transfer scores and the language similarity measures. The models were fine-tuned by using PyTorch and the Huggingface Transformers library [175]. The hardware used was an Nvidia GTX 1080Ti GPU.

5.2.2 Results

Both of the multilingual transformer models (mBERT, XLM-R) were fine-tuned with all of the proposed languages for each task (sentiment analysis, NER, DEP) introduced earlier. The models were fine-tuned using only the training dataset from a single language before the evaluation step. The model evaluation scores are presented in Tables 5.4 and 5.5 for sentiment analysis, Tables 5.6 and 5.7 for NER and Tables 5.8 and 5.9 for DEP.

Table 5.4: Sentiment analysis: F1-scores for mBERT

		TARGET							
		Danish	**English**	**German**	**Croatian**	**Polish**	**Russian**	**Japanese**	**Korean**
	Danish	0.976	0.875	0.951	0.941	0.934	0.876	0.800	0.881
	English	0.942	0.935	0.935	0.921	0.921	0.849	0.645	0.838
	German	0.901	0.816	0.971	0.908	0.889	0.828	0.711	0.741
SOURCE	**Croatian**	0.952	0.883	0.948	0.973	0.940	0.863	0.802	0.881
	Polish	0.952	0.876	0.948	0.948	0.967	0.861	0.771	0.878
	Russian	0.949	0.862	0.939	0.938	0.933	0.957	0.774	0.867
	Japanese	0.908	0.799	0.903	0.894	0.870	0.807	0.914	0.869
	Korean	0.940	0.848	0.935	0.930	0.909	0.850	0.815	0.957

Table 5.5: Sentiment analysis: F1-scores for XLM-R

		TARGET							
		Danish	**English**	**German**	**Croatian**	**Polish**	**Russian**	**Japanese**	**Korean**
	Danish	0.972	0.857	0.932	0.934	0.926	0.869	0.749	0.846
	English	0.939	0.925	0.920	0.922	0.916	0.844	0.705	0.816
	German	0.882	0.791	0.966	0.890	0.871	0.816	0.671	0.711
SOURCE	**Croatian**	0.945	0.859	0.925	0.969	0.929	0.876	0.763	0.835
	Polish	0.946	0.853	0.929	0.939	0.960	0.865	0.632	0.834
	Russian	0.918	0.792	0.895	0.913	0.901	0.953	0.726	0.832
	Japanese	0.888	0.793	0.880	0.871	0.851	0.773	0.905	0.832
	Korean	0.921	0.804	0.903	0.911	0.880	0.824	0.690	0.953

Looking at the results, it can clearly be said that XLM-R outperformed mBERT in all of the tasks. The only exception to this was the sentiment analysis task,

Table 5.6: NER: F1-scores for mBERT

		TARGET							
		Danish	**English**	**German**	**Croatian**	**Polish**	**Russian**	**Japanese**	**Korean**
SOURCE	**Danish**	0.957	0.813	0.801	0.480	0.763	0.791	0.675	0.640
	English	0.778	0.930	0.827	0.744	0.852	0.729	0.773	0.652
	German	0.770	0.866	0.936	0.751	0.879	0.805	0.766	0.648
	Croatian	0.667	0.710	0.748	0.876	0.727	0.770	0.707	0.622
	Polish	0.691	0.676	0.702	0.695	0.956	0.648	0.754	0.625
	Russian	0.759	0.825	0.764	0.761	0.867	0.946	0.759	0.564
	Japanese	0.754	0.827	0.761	0.659	0.693	0.743	0.926	0.673
	Korean	0.602	0.694	0.668	0.670	0.675	0.705	0.700	0.867

Table 5.7: NER: F1-scores for XLM-R

		TARGET							
		Danish	**English**	**German**	**Croatian**	**Polish**	**Russian**	**Japanese**	**Korean**
SOURCE	**Danish**	0.975	0.868	0.876	0.723	0.958	0.910	0.873	0.755
	English	0.955	0.941	0.941	0.820	0.961	0.934	0.920	0.781
	German	0.960	0.926	0.948	0.846	0.975	0.930	0.918	0.765
	Croatian	0.920	0.842	0.872	0.911	0.919	0.889	0.834	0.723
	Polish	0.949	0.885	0.897	0.877	0.981	0.897	0.891	0.740
	Russian	0.923	0.908	0.915	0.611	0.951	0.951	0.880	0.737
	Japanese	0.955	0.909	0.920	0.847	0.961	0.926	0.936	0.789
	Korean	0.827	0.752	0.799	0.491	0.751	0.820	0.840	0.900

Table 5.8: DEP: LAS-scores for mBERT

		TARGET							
		Danish	**English**	**German**	**Croatian**	**Polish**	**Russian**	**Japanese**	**Korean**
SOURCE	**Danish**	0.860	0.545	0.631	0.619	0.556	0.647	0.092	0.026
	English	0.652	0.891	0.670	0.624	0.570	0.653	0.165	0.021
	German	0.635	0.603	0.842	0.672	0.613	0.733	0.130	0.062
	Croatian	0.581	0.607	0.633	0.893	0.645	0.778	0.124	0.030
	Polish	0.520	0.518	0.577	0.676	0.924	0.760	0.112	0.023
	Russian	0.594	0.604	0.643	0.730	0.666	0.878	0.131	0.020
	Japanese	0.132	0.148	0.163	0.114	0.117	0.126	0.926	0.033
	Korean	0.058	0.065	0.060	0.035	0.045	0.054	0.035	0.293

Table 5.9: DEP: LAS-scores for XLM-R

		TARGET							
		Danish	**English**	**German**	**Croatian**	**Polish**	**Russian**	**Japanese**	**Korean**
SOURCE	**Danish**	0.888	0.679	0.725	0.706	0.672	0.715	0.095	0.366
	English	0.733	0.911	0.728	0.720	0.700	0.716	0.112	0.364
	German	0.712	0.681	0.854	0.751	0.732	0.784	0.066	0.405
	Croatian	0.639	0.668	0.702	0.910	0.798	0.818	0.069	0.375
	Polish	0.614	0.603	0.676	0.780	0.945	0.804	0.049	0.384
	Russian	0.642	0.645	0.722	0.801	0.796	0.890	0.101	0.378
	Japanese	0.118	0.132	0.172	0.098	0.122	0.106	0.937	0.317
	Korean	0.326	0.292	0.381	0.298	0.325	0.304	0.183	0.877

where mBERT slightly outperformed XLM-R. It can be noted from the results that the highest transfer scores usually belong to the languages in the same language family as the source language (English, German, Danish - Germanic; Croatian, Polish, Russian - Slavic; Japanese, Korean - Koreano-Japonic). Also, most of the time there is a clear difference in the scores when evaluating with the same language as the source compared to zero-shot cross-lingual transfer. The exceptions to this are the sentiment analysis task for both models and the NER task for XLM-R.

In dependency parsing, XLM-R slightly outperformed mBERT as expected. However, in the sentiment analysis task mBERT scored slightly higher than XLM-R overall, with both models scoring high across all language pairs. Some language pairs even achieving zero-shot cross-lingual transfer F-score of over 0.95. In this task, there seems not to be a clear pattern what kind of language pairs tend to yield higher performance. For example, Slavic languages seem to work better as sources for Danish compared to German languages in the case of both models. The scores are also similarly high across the board for the NER task with XLM-R, with the model being able to achieve very high scores with zero-shot transfer. The performance difference between mBERT and XLM-R is also more noticeable in the case of NER.

As can be seen from Table 5.10, both Japanese and Korean worked decently well as cross-lingual transfer sources for both sentiment analysis and NER tasks, besides being very different from the other languages used in the experiments as they are the only non Indo-European languages. However, in the case of DEP their performance is extremely low. Except for this case with DEP, all of the proposed languages seem to be quite equal as cross-lingual transfer sources in general. Interestingly, German, Croatian and Russian seem to perform slightly better overall compared to the other languages, especially with mBERT. A similar phenomenon was also experienced by Turc et al. [94] and in the previous Chapter with cyberbullying detection.

5.2.3 Effect of Linguistic Similarity

I calculated the correlation between the zero-shot cross-lingual transfer results of the two models and each of the three proposed linguistic similarity metrics (EzGlot, eLinguistics, WALS) in all proposed NLP tasks using both Pearson's and

Table 5.10: Average scores for each source language on each task

	mBERT			XLM-R		
	Sentiment	**NER**	**DEP**	**Sentiment**	**NER**	**DEP**
Danish	0.904	0.740	0.497	0.886	0.867	0.606
English	0.873	0.786	0.531	0.874	0.907	0.623
German	0.845	0.803	0.536	0.825	0.909	0.623
Croatian	0.905	0.728	0.536	0.888	0.864	0.622
Polish	0.900	0.719	0.514	0.870	0.890	0.607
Russian	0.902	0.781	0.533	0.866	0.860	0.622
Japanese	0.871	0.754	0.220	0.849	0.905	0.250
Korean	0.898	0.698	0.081	0.861	0.773	0.373

Spearman's correlation coefficients (ρ-value). I ignored some of the language pairs when calculating the correlations with the EzGlot metric as some of the similarity values were missing. The correlation analysis results are shown in Table 5.11 for sentiment analysis, Table 5.12 for NER, Table 5.13 for DEP.

Table 5.11: Sentiment analysis: Pearson's and Spearman's correlation coefficients for model F1 scores and linguistic similarity metrics

	Pearson				Spearman			
	XLM-R		mBERT		XLM-R		mBERT	
	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**
WALS	-0.297	0.017	-0.645	0.000	-0.331	0.008	-0.537	0.000
EzGlot	0.389	0.005	0.729	0.000	0.533	0.000	0.586	0.000
eLinguistics	-0.355	0.004	-0.648	0.000	-0.413	0.001	-0.652	0.000

Table 5.12: NER: Pearson's and Spearman's correlation coefficients for model F1 scores and linguistic similarity metrics

	Pearson				Spearman			
	XLM-R		mBERT		XLM-R		mBERT	
	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**
WALS	-0.514	0.000	-0.500	0.000	-0.510	0.000	-0.486	0.000
EzGlot	0.494	0.000	0.427	0.002	0.464	0.001	0.401	0.004
eLinguistics	-0.580	0.000	-0.517	0.000	-0.608	0.000	-0.553	0.000

Table 5.13: DEP: Pearson's and Spearman's correlation coefficients for model LAS scores and linguistic similarity metrics

	Pearson				**Spearman**			
	XLM-R		**mBERT**		**XLM-R**		**mBERT**	
	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**
WALS	-0.781	0.000	-0.718	0.000	-0.844	0.000	-0.693	0.000
EzGlot	0.588	0.000	0.516	0.000	0.694	0.000	0.561	0.000
eLinguistics	-0.845	0.000	-0.840	0.000	-0.897	0.000	-0.867	0.000

Looking at the results, one can say that for both Pearson's and Spearman's correlation, the results show that there is mostly a strong correlation between WALS and eLinguistics metrics and the cross-lingual zero-shot transfer score and a strong-moderate correlation between the EzGlot metric and the transfer scores for all of the tasks except sentiment analysis where the correlation is noticeably lower for XLM-R, staying at a moderate level with all of the linguistic similarity metrics.

The strongest correlations are found in the dependency parsing task with XLM-R, with the highest absolute Spearman's correlation being 0.897 with eLinguistics metric. The second strongest correlation is found in NER. The sentiment analysis task has the weakest correlations overall. Also, the results show that p-value $<$ 0.05 for all of the tasks, models and metrics, indicating statistical significance. For both of the models, the correlation for WALS and eLinguistics metrics are generally higher than EzGlot, except in the case of sentiment analysis, where EzGlot's correlation is slightly higher for both Pearson and Spearman. Also, the correlations were generally slightly stronger with mBERT in sentiment analysis, while XLM-R had higher correlations in both NER and DEP tasks.

However, the results changed drastically for all tasks except dependency parsing, when I removed the anchor points of same source-target language pairs (monolingual scenarios), leaving only the zero-shot transfer results. This was necessary to do in order remove the bias brought by the monolingual data points, as the scores are higher and the languages would also be the most similar (same). The results after removing the same source-target language pairs are shown in Table 5.14 for sentiment analysis, Table 5.15 for NER, Table 5.16 for DEP.

First of all, for both eLinguistics and WALS similarity metrics, the correlations generally dropped from strong to moderate for the NER task, while EzGlot's

correlation fell close to zero in both of these tasks and also lost all statistical significance. Interestingly, for sentiment analysis, the result looks completely opposite. EzGlot's metric only fell slightly while WALS' and eLinguistics' correlation plummeted down and lost statistical significance for XLM-R in this task. The correlations in the dependency parsing task only dropped slightly for all of the linguistic similarity metrics. Also, after removing the same source-target language pairs, the strongest correlations are still found in DEP, followed by NER, while sentiment analysis still has the weakest correlations.

Table 5.14: Sentiment analysis: Pearson's and Spearman's correlation coefficients for model F1 scores and linguistic similarity metrics for zero-shot only

	Pearson				**Spearman**			
	XLM-R		**mBERT**		**XLM-R**		**mBERT**	
	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**
WALS	-0.111	0.415	-0.262	0.051	-0.168	0.216	-0.315	0.018
EzGlot	0.327	0.035	0.303	0.051	0.403	0.008	0.313	0.044
eLinguistics	-0.229	0.090	-0.392	0.003	-0.284	0.034	-0.487	0.000

Table 5.15: NER: Pearson's and Spearman's correlation coefficients for model F1 scores and linguistic similarity metrics for zero-shot only

	Pearson				**Spearman**			
	XLM-R		**mBERT**		**XLM-R**		**mBERT**	
	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**
WALS	-0.336	0.011	-0.347	0.009	-0.316	0.017	-0.293	0.028
EzGlot	0.173	0.274	0.120	0.448	0.170	0.282	0.095	0.549
eLinguistics	-0.453	0.000	-0.384	0.003	-0.458	0.000	-0.389	0.003

5.3 Discussion

5.3.1 Transfer Language Performance

XLM-R outperforming mBERT generally matches the expectations, as it also did so on a variety of benchmark tasks [90, 91]. The reason behind this most likely is the fact that XLM-R uses a vastly larger amount of data for pretraining

Table 5.16: DEP: Pearson's and Spearman's correlation coefficients for model LAS scores and linguistic similarity metrics for zero-shot only

	Pearson				**Spearman**			
	XLM-R		**mBERT**		**XLM-R**		**mBERT**	
	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**
WALS	-0.738	0.000	-0.661	0.000	-0.769	0.000	-0.588	0.000
EzGlot	0.421	0.005	0.373	0.015	0.488	0.001	0.349	0.024
eLinguistics	-0.795	0.000	-0.822	0.000	-0.848	0.000	-0.842	0.000

compared to mBERT. The performance difference between the two models is the most clear in the NER task.

According to the results, simply choosing English as the transfer source did not yield top results most of the time, sometimes even as a source language to other languages in the Germanic language group. For example, it had a lower than average performance in sentiment analysis and an average performance in the other two tasks probably due to its simplicity when compared to both Danish and German. It was also slightly outperformed by Slavic languages in some cases when used as a source for other Germanic languages. Another reason could be the influence of French [176, 177, 178], which might further distance it from the other Germanic languages. Also, the differences in morphology could be a factor here. Danish and German probably work better with each other due to a great amount of historic mutual influence.

In sentiment analysis, the models achieved slightly better scores with English and it was on-par with other Germanic languages. However, all of the Slavic languages still tended to work slightly better as transfer sources. These results show that other languages should also be considered over English as the cross-lingual transfer source if available. For the other two tasks, NER and DEP, English performed well and showed to be a good transfer source for the other two Germanic languages.

In most cases, using languages from the same language family as the source language yielded the highest cross-lingual transfer scores. This matches with the typical intuition-based selection process used to select source language for cross-lingual transfer. However, relying only on intuition and looking purely at language families when when selecting the transfer language will lead to diminished results

in some cases.

One example would be taking Polish as the target language for DEP task. One could expect that in this case, the best transfer languages would be Croatian and Russian, but looking at the results (Tables 5.6 and 5.7) German had a higher cross-lingual transfer score even though it is from the Germanic language family, not Slavic. This could be, for example, due to mutual influence of these two languages. The grammar of both Danish and English is relatively simple compared to German, which could aid them in generalizing better with one another. Looking at the scores, it can be noted that German is a good source for both Germanic and Slavic languages, which could mean that the historical mutual influence between the Germans and Slavs could be a factor here. Furthermore, German, in addition to having a higher average performance on most tasks, tended to also work exceptionally well also as a source language for other Slavic languages, most likely because of the reasons discussed above.

In addition, Japanese and Korean did not achieve good scores with one another, contrary to the expectations, and were even slightly outperformed by other languages approximately half of the time, even though being more similar with each other compared to any of the other proposed languages. Here the reason could be, for example, the differences in the writing systems, as neither of these two languages use alphabets and their systems also greatly differ from each other.

Also, both Russian and Croatian had a higher than average performance on most of the tasks. This was similar to the previous Chapter where Russian performed exceptionally well as a transfer source for offensive language identification. However, unlike in the cyberbullying task, Russian did not perform noticeably well as the transfer language source for Korean and Japanese. Thus the phenomenon experienced previously is most likely related to the topic of offensive language identification itself or to the properties of these specific datasets. I will investigate this in later research. Also, Japanese and Korean had a satisfying performance as source languages for most of the tasks, even though they are fundamentally different from all of the other languages used, as they are the only non Indo-European languages in the proposed set. This demonstrates that multilingual transformer models are also able to leverage knowledge even from very distant languages.

5.3.2 Analysis of Linguistic Similarity Metrics

The correlation between cross-lingual transfer performance and the similarity metrics were strong or moderate with all of the proposed metrics, which would suggest that using even a single feature such as lexical information or by comparing phonetic consonants is still effective to some extent. However, when considering only the zero-shot transfer results, EzGlot's similarity metric's correlation dropped drastically and out of statistical significance in the NER task. This shows that it does not necessarily rely on lexical features and that other linguistic features need to be considered when choosing the source language for NER. On the other hand, the opposite happened in the sentiment analysis task, with both WALS and eLinguistics metrics' correlation dropping drastically and out of statistical significance. This hints the importance of lexical similarity when choosing the source language for sentiment analysis tasks.

Surprisingly, even though using only a predefined set of phonetic consonants for its calculation, the correlation of eLinguistics' similarity metric was stronger in all tasks compared to the the correlation of the WALS metric, which I quantified from the WALS database using linguistic features from multiple domains. The reason behind this could be that including all of the common features between each language pair could have caused too many irrelevant features to be included. This can cause a possible bias the metric calculation. In the future, I will take another glance at the WALS database, aiming for a better quantification by looking at the importance of each feature group (syntactic, lexical, phonetic, etc.) [101] and weighing accordingly while filtering out redundant features in order to develop an even more effective and comprehensive similarity metric.

However, as shown in the previous Chapter, the eLinguistics metric also has its weak points as it is based on only a single aspect of language, even though its correlation being the strongest. One can see from Table 5.1 that eLinguistics shows Japanese being very distant from Korean, being at the same level as Polish and Russian, with Croatian being seemingly closer to Korean than Japanese, which is not true due to the similarities in the vocabulary and grammar of Japanese and Korean. Taking a look at Table 5.3, it is clear that the WALS metric is a lot more robust to this kind of errors. The reason most likely is that instead of using only a single linguistic feature like the eLinguistics metric, the WALS metric is based on a

large amount of features spanning over multiple domains.

5.3.3 Task-Specific Analysis

Looking at the results of the sentiment analysis task, it is clear that the results are very high across the board and the score differences between language groups are also a lot smaller, with sometimes languages from other language groups than the target emerging as the best performers. This is the case for example with Danish, as Croatian achieved the highest zero-shot transfer scores for both mBERT and XLM-R instead of another Germanic language.

A trait only observed in this task was that EzGlot was the only metric keeping the correlation in the zero-shot setting, hinting the importance of lexical features. One could argue that the reason behind the overall high scores might be due to the task being too easy, as it simply required the classification of the entries into positive and negative. This has also been shown in other research [179]. However, this also shows that it could be possible to achieve at least close to state-of-the-art results with multilingual transformer models in a zero-shot cross-lingual setting. This raises questions about how to improve the cross-lingual models to better utilize cross-lingual transfer. In the future, it would be useful to further investigate the models' behaviour in zero-shot setting. This could also be useful in the further development of measures to support low-resource languages.

For mBERT, the zero-shot results of the NER task look clearly lower than with same language pairs and quite even across all of the proposed languages and the languages belonging to the same group having generally a slightly higher score. However, the results of XLM-R closely resemble those of the sentiment analysis task as the results are high across all language pairs. This further shows the potential these models have in relieving the issues with low-resource languages. Also, there is a moderate correlation between the zero-shot transfer performance and linguistic similarity for both WALS and eLinguistics metrics, meaning they can be used in the transfer language selection process.

The results in DEP also seem clearly lower than with same language pairs and even across the board. The languages belonging to the same group also generally have a slightly higher score. As the task requires the understanding of syntax and grammar and the scores are still reasonably high overall, the results also support

the studies claiming that the cross-lingual transformer models are able to learn grammar without explicit information [180].

On the other hand, Japanese and Korean, unlike in previous tasks, had very poor performance as source languages while also being very difficult target languages. This could mean that the model is unable to generalize to the syntax and grammar of Indo-European languages with Japonic-Koreanic languages and vice versa. The reason might be due to the differences in writing systems. Here, both models keep a strong correlation between the zero-shot transfer performance and linguistic similarity for both WALS and eLinguistics metrics in a zero-shot setting, meaning they can be used in the transfer language selection process.

As both of the eLinguistics and WALS similarity metrics correlated with transfer language performance for NER and DEP, they can be used for the cross-lingual transfer language selection process, at least for these tasks. In addition, a metric focusing on lexical features (EzGlot) could be used for sentiment analysis. Based on these results, it is better to look for high-resource languages that have proper data available and are as close as possible to the target language based on these similarity metrics instead of making a decision based on intuition or simply blindly choosing any language from the same language family. This allows one to make a more informed and effective decision and makes model development more efficient.

5.3.4 Future Research

In the future, I am planning to analyze, what kind of linguistic features are the most important from the point of view of cross-lingual transfer. A solution could be grouping the features presented in WALS into syntactic, lexical, phonetic, etc., and calculating, which feature group has the strongest correlation with the cross-lingual transfer performance. I could then re-quantify the WALS database using this information in order to develop an even more effective and comprehensive similarity metric. It would also be beneficial if the WALS project received more attention and the feature matrix became more densely populated.

As shown by the DEP task, the models might be able to learn syntax and grammar without any explicit information. This could mean that adding explicit syntactic and grammatical information to the pre-training process of the models might also improve their performance. I will take a look at this in the future. Also,

as the models achieved zero-shot transfer scores rivaling those of the monolingual settings, especially in sentiment analysis, it would be useful to perform an in-depth investigation about the models' behaviour in a zero-shot transfer learning setting to possibly find insights on how to improve their transfer learning capabilities.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, I presented the results in improving automatic cyberbullying detection with Feature Density and Cross-Lingual Zero-Shot transfer. I showed that in order to hasten the development of detection systems, it is possible to both optimize the model training process using Feature Density and to train models for languages lacking data by leveraging knowledge from more high-resource ones.

The main contributions of this thesis can be summarized as follows:

1. Feature Density can be utilized to reduce of the number of required experiments iterations.

2. Dataset complexity cannot be measured with Feature Density alone.

3. Linguistic preprocessing can improve classifier performance.

4. Zero-shot cross-lingual transfer has potential in achieving results close to that of monolingual settings.

5. A suitable transfer language can be found by using Linguistic similarity for many NLP tasks.

6. Selecting a transfer language based on intuition or simply by language family often results in performance.

7. The linguistic similarity metric quantified from World Atlas of Language Structures is comparably robust.

I began this thesis with an overview of previous research, including research on cyberbullying detection, model training efficiency and cross-lingual transfer (Chapter 2).

In Chapter 3, I presented the research on Feature Density and linguistically-backed preprocessing methods, applied in cyberbullying detection. Both concepts are relatively novel to the field. I studied the effectiveness of Feature Density using a variety of linguistically-backed feature preprocessing methods to estimate dataset complexity and classifier performance.

From the results I concluded that most of the classifier performances, excluding CNNs, had a strong negative correlation with FD, which means weaker performance if a lot of linguistic information is added. Depending on the dataset and classifier, the best performance was in most cases between 50-200% of the base FD (TOK). For CNNs, I was unable confirm the positive correlation between classifier performance and FD for the Japanese dataset as there was usually only a very weak positive or no correlation between FD and the classifier performance. Still, there could be potential in the higher FD preprocessing types for CNNs. I also discovered that the best results for English were usually obtained by slightly increasing the base feature density, while for the Japanese and Polish dataset it was the opposite. The reason behind this is probably due to the difference in language complexity of English compared to Japanese and Polish.

The results indicated that natural language dataset complexity cannot be measured by using Lexical Density or Feature Density alone, even if it could be used to measure the complexity of the language itself. However, I found out that the relative change in Feature Density when lemmatizing each dataset matches the ranking of the dataset scores. I also noticed that even with datasets of similar sizes and topics, but with different languages, there can be huge differences in classification results. This means it is not enough to develop the tools and models with just English in mind, as it is a much less complex language when compared to others like Japanese or Polish. Which means state-of-the-art results achieved for English are not representative globally for the task in question, but rather locally, for the task done in this particular language.

Using tokens worked slightly better for English while lemmas were better for Japanese and especially Polish. The reason being most likely related to the complexity of the language, as the base feature density for tokens is too high in Japanese and Polish for classifiers to correctly generalize on the dataset of this size and thus lowering the complexity by lemmatizing resulted in an increased performance. Chunking and dependencies generally had a poor performance but dependencies still showed some potential with neural networks. From the supplementary preprocesing methods, stopwords and POS tagging generally resulted in positive results, except for the questionable use of stopwords with the Japanese dataset. NER showed mixed results, being usable mainly with Japanese while alphabetic filtering had poor results with all of the datasets due to non-alphabetic tokens also carrying useful information, at least in the context of cyberbullying detection. In general, the preprocessing had a positive effect on the scores which proves that using linguistically-backed preprocessing can be used to increase classifier performance, at least in the context of cyberbullying. I also discovered that parts of speech information can be used with tokens or lemmas to produce the most stable and high performance feature sets.

I proposed that the method can be applied by first discarding the FD ranges where the overall weakest feature sets are (POS, CHNK, etc.). Then running a small subset of the experiments with a set interval between preprocessing type feature densities and iterate around the most probable peak to find the maximum performance. I concluded that the method could save as much as driving a new car for almost 50 kilometers in greenhouse gas emissions when training a simple CNN model with the English dataset by discarding the weakest feature sets. I assume that the method could also be applied to more modern models than those used in this study.

In Chapter 4, I studied the selection of transfer languages for automatic cyberbullying detection. I demonstrated the effectiveness of cross-lingual transfer learning for zero-shot offensive language identification on a target language. This way it is possible to leverage existing data from higher-resource languages in order to improve the performance of languages lacking proper data. I showed that there is a strong correlation between the proposed linguistic similarity metrics and the cross-lingual transfer performance. As the languages get more distant, the transfer

performance decreases. This makes it possible to choose an optimal transfer language by comparing the similarity of languages instead of relying on intuition. As shown by the experiments, choosing languages from the same language family is not always the best option. Instead, one should use different linguistic features to compare the languages before selection and base the choice on a linguistic similarity metric instead. The experiments also showed that lexical information alone is not enough to determine the optimal transfer languages.

I also showed that it is possible to achieve good performance on the target language in a zero-shot cross-lingual transfer setting. This helps in developing better detection systems for offensive language identification, especially when dealing with low-resource languages. This is particularly important because of the severity of the problem and the fact that social media is used in thousands of languages, of which only a small fraction even have proper data to train the detection models on.

Lastly, I developed a novel linguistic similarity metric consisting of various linguistic features by using the WALS database. The proposed method did not show the strongest correlation with the transfer performance, but it still showed potential as a metric that could be useful for the selection process, especially if given a more refined or inclusive feature set. In the future, I will aim for a better quantification of the WALS database in order to develop an even more effective and comprehensive linguistic similarity metric.

The proposed method for cross-lingual transfer language selection could also be useful as a general method for other Natural Language Processing tasks, not only for harmful online content detection. This was explored in Chapter 5.

In Chapter 5, I studied generalizability of the method developed in the previous Chapter using three different NLP tasks, namely, sentiment analysis, NER, and dependency parsing. I showed the effectiveness of cross-lingual zero-shot transfer learning with a total of eight languages from three language families. In this way, existing data from higher-resource languages may be used to improve the performance of languages that lack sufficient data also for other tasks than cyberbullying detection.

I found a strong correlation between the suggested linguistic similarity metrics and cross-lingual transfer performance. This shows that the proposed method for cross-lingual transfer language selection could also be useful as a general method

for other Natural Language Processing tasks, at least based on the proposed tasks. The experiments also demonstrated that lexical information alone is insufficient to determine the optimal transfer languages at least for the tasks of NER and DEP. I also showed that it is possible to achieve good performance on the target language in a zero-shot cross-lingual transfer setting with multiple NLP tasks. This helps in developing better systems, especially when dealing with low-resource languages.

I improved a novel linguistic similarity metric consisting of various linguistic features by using the WALS database. However, the proposed method still did not show the strongest correlation with the transfer performance. In the future, I will reassess the importances of the linguistic features used in the similarity metric calculation in order to have a more refined feature set, aiming to create an even more effective and comprehensive linguistic similarity metric.

Lastly, even though the overall high scores in the sentiment analysis task might be caused by the task being too easy, it also shows that it could be possible to achieve results close to those of a monolingual fine-tuning in a zero-shot cross-lingual transfer setting. This means it could be useful to thoroughly investigate the models' behaviour in zero-shot setting in order to find insights to improving their transfer capabilities. Also, as the DEP task demonstrated that the models might have a capability to understand grammar, adding explicit syntactic and grammatical information to the models' pre-training could also increase performance.

Ultimately, this research enables faster and more efficient development of cyberbullying detection systems by reducing the number of required experiment iterations and increasing model performance. Furthermore, the research makes it possible to create systems for languages lacking data without having to go through the costly process of collecting and annotating datasets.

6.2 Future Work

The usage of high Feature Density preprocessings, namely dependency parsing, showed potential with CNNs with English and Japanese datasets and needs to be confirmed with larger datasets in the future. Also, more exact ideal feature densities need to be confirmed for each classifier and language by using datasets of different sizes to make the ranking of classifiers by FD as accurate as possible.

Due to the linguistic complexity of the Polish language, more data would most likely be required to make the use of neural networks viable. The effect of FD with neural networks in Polish needs to be explored further using a larger dataset in the future.

In order to use FD as a measure to estimate the complexity of a dataset, a more in depth study with more datasets of different sizes, topics and languages is required. Even if the complexity of languages cannot be perfectly quantified by a simple measure like Feature Density, there could be other measures or a combination of them that might be useful in better ranking datasets in terms of complexity.

Linguistic preprocessings had a positive effect on the scores, which shows their potential as a method for increasing classifier performance. In the future, pretraining the linguistic embeddings on a larger corpus need to be experimented on instead of relying on a small dataset. The effects of linguistic preprocessing should be also confirmed with other models, such as recurrent neural networks and the state-of-the-art pretrained language models like BERT. Also, the plan is to do a qualitative evaluation on the different kinds of linguistic embeddings in order to analyze their effects more deeply. Also I am going to train the embeddings on larger datasets and measure the classification performance on other languages to confirm and further explore the results of this study.

Furthermore, the plan is to analyze, what kind of linguistic features are the most important from the point of view of cross-lingual transfer. A solution could be grouping the features presented in WALS into syntactic, lexical, phonetic, etc., and calculating, which feature group has the strongest correlation with the cross-lingual transfer performance. It could be possible to then re-quantify the WALS database using this information in order to develop an even more effective and comprehensive similarity metric. It would also be beneficial if the WALS project received more attention and the feature matrix became more densely populated.

As shown by the DEP task, the models might be able to learn syntax and grammar without any explicit information. This could mean that adding explicit syntactic and grammatical information to the pre-training process of the models might also improve their performance. I will take a look at this in the future. Also, as the models achieved zero-shot transfer scores rivaling those of the monolingual settings, especially in sentiment analysis, it would be useful to perform an in-depth

investigation about the models' behaviour in a zero-shot transfer learning setting to possibly find insights on how to improve their transfer learning capabilities.

Appendix A

F-scores and Standard Errors of Classifier-Preprocessing Pairs

Table A.1: English dataset: F-scores and average standard error for classifier-preprocessing pairs (F1±stderr)

	LBFGS LR	NewtonLR	LinearSVM	SGD SVM	KNN	NaiveBayes	RandForest
CHNK	0.727±0.234	0.726±0.242	0.718±0.257	0.736±0.237	0.570±0.390	0.674±0.247	0.613±0.366
CHNKALPHA	0.684±0.262	0.681±0.259	0.669±0.266	0.683±0.263	0.607±0.339	0.643±0.263	0.647±0.324
CHNKNER	0.718±0.238	0.723±0.236	0.721±0.248	0.737±0.234	0.582±0.385	0.669±0.248	0.603±0.379
CHNKNERALPHA	0.675±0.262	0.676±0.260	0.663±0.271	0.663±0.267	0.599±0.341	0.641±0.262	0.618±0.336
CHNKNERR	0.688±0.257	0.695±0.258	0.702±0.265	0.699±0.257	0.580±0.380	0.653±0.255	0.603±0.368
CHNKNERRALPHA	0.660±0.269	0.663±0.268	0.651±0.277	0.657±0.276	0.603±0.335	0.626±0.267	0.616±0.343
CHNKNERRSTOP	0.686±0.265	0.684±0.267	0.684±0.274	0.694±0.264	0.577±0.378	0.629±0.259	0.635±0.325
CHNKNERRSTOPALPHA	0.618±0.280	0.617±0.282	0.591±0.288	0.607±0.287	0.404±0.233	0.598±0.276	0.620±0.304
CHNKNERSTOP	0.724±0.241	0.724±0.241	0.715±0.254	0.724±0.241	0.582±0.389	0.663±0.251	0.635±0.339
CHNKNERSTOPALPHA	0.666±0.261	0.661±0.265	0.644±0.279	0.668±0.260	0.386±0.218	0.615±0.268	0.659±0.301
CHNKSTOP	0.722±0.241	0.721±0.239	0.711±0.256	0.723±0.239	0.577±0.379	0.670±0.247	0.667±0.308
CHNKSTOPALPHA	0.629±0.277	0.637±0.274	0.606±0.286	0.619±0.277	0.395±0.226	0.608±0.271	0.649±0.305
DEP	0.617±0.354	0.619±0.349	0.568±0.398	0.587±0.381	0.243±0.094	0.617±0.280	0.536±0.435
DEPALPHA	0.609±0.349	0.612±0.346	0.588±0.372	0.601±0.356	0.314±0.156	0.600±0.288	0.545±0.416
DEPNER	0.624±0.339	0.621±0.341	0.574±0.391	0.585±0.379	0.242±0.092	0.611±0.279	0.528±0.443
DEPNERALPHA	0.585±0.308	0.589±0.311	0.561±0.317	0.579±0.317	0.213±0.069	0.607±0.282	0.578±0.384
DEPNERR	0.610±0.350	0.614±0.341	0.571±0.389	0.587±0.381	0.241±0.094	0.611±0.283	0.533±0.426
DEPNERRALPHA	0.606±0.352	0.605±0.355	0.589±0.367	0.602±0.359	0.312±0.156	0.596±0.293	0.537±0.418
DEPNERRSTOP	0.602±0.358	0.599±0.360	0.564±0.395	0.568±0.393	0.273±0.125	0.615±0.291	0.543±0.421
DEPNERRSTOPALPHA	0.584±0.366	0.584±0.365	0.560±0.386	0.581±0.376	0.386±0.223	0.599±0.313	0.544±0.421
DEPNERSTOP	0.611±0.348	0.602±0.352	0.564±0.397	0.576±0.386	0.274±0.119	0.604±0.284	0.527±0.438
DEPNERSTOPALPHA	0.535±0.313	0.531±0.314	0.523±0.319	0.523±0.310	0.297±0.151	0.543±0.287	0.563±0.372
DEPSTOP	0.606±0.365	0.595±0.367	0.562±0.400	0.571±0.399	0.276±0.122	0.616±0.288	0.544±0.426
DEPSTOPALPHA	0.586±0.371	0.587±0.365	0.564±0.386	0.588±0.387	0.388±0.221	0.594±0.309	0.539±0.415
LEM	0.781±0.180	0.786±0.187	0.784±0.190	0.790±0.182	0.634±0.326	0.715±0.221	0.724±0.253
LEMALPHA	0.755±0.195	0.764±0.198	0.745±0.204	0.765±0.198	0.294±0.179	0.703±0.225	0.718±0.244
LEMNER	0.784±0.187	0.782±0.183	0.787±0.182	0.792±0.176	0.631±0.330	0.710±0.224	0.716±0.261
LEMNERALPHA	0.763±0.197	0.764±0.196	0.765±0.203	0.767±0.194	0.637±0.315	0.699±0.227	0.710±0.255
LEMNERR	0.740±0.217	0.737±0.219	0.742±0.217	0.740±0.214	0.601±0.335	0.692±0.234	0.697±0.280
LEMNERRALPHA	0.729±0.222	0.728±0.222	0.725±0.227	0.725±0.222	0.614±0.321	0.685±0.237	0.699±0.282
LEMNERRSTOP	0.737±0.220	0.734±0.221	0.726±0.228	0.732±0.220	0.609±0.329	0.682±0.235	0.727±0.245
LEMNERRSTOPALPHA	0.732±0.222	0.732±0.225	0.714±0.236	0.727±0.227	0.624±0.323	0.674±0.239	0.723±0.249
LEMNERSTOP	0.782±0.185	0.783±0.187	0.782±0.186	0.792±0.179	0.634±0.330	0.706±0.225	0.745±0.230
LEMNERSTOPALPHA	0.770±0.195	0.767±0.195	0.752±0.201	0.767±0.200	0.640±0.304	0.693±0.226	0.739±0.231
LEMPOS	0.778±0.204	0.778±0.198	0.788±0.202	0.790±0.202	0.517±0.365	0.711±0.222	0.663±0.310
LEMPOSALPHA	0.768±0.204	0.772±0.210	0.772±0.211	0.768±0.209	0.522±0.210	0.700±0.228	0.654±0.298
LEMPOSS	0.764±0.202	0.765±0.207	0.769±0.205	0.767±0.202	0.564±0.359	0.713±0.227	0.658±0.270
LEMPOSSALPHA	0.760±0.200	0.758±0.204	0.753±0.213	0.758±0.209	0.406±0.254	0.705±0.230	0.669±0.259
LEMPOSSSTOP	0.763±0.189	0.766±0.187	0.767±0.186	0.774±0.177	0.566±0.296	0.709±0.219	0.706±0.327
LEMPOSSSTOPALPHA	0.762±0.198	0.766±0.193	0.748±0.194	0.765±0.190	0.490±0.297	0.702±0.227	0.713±0.314
LEMPOSSTOP	0.780±0.187	0.781±0.191	0.788±0.183	0.788±0.186	0.642±0.285	0.708±0.222	0.708±0.261
LEMPOSSTOPALPHA	0.770±0.193	0.769±0.195	0.766±0.202	0.768±0.191	0.669±0.282	0.696±0.227	0.718±0.256
LEMSTOP	0.787±0.183	0.786±0.185	0.784±0.192	0.791±0.181	0.641±0.320	0.713±0.221	0.754±0.229

LEMSTOPALPHA	0.772±0.190	0.766±0.192	0.766±0.196	0.773±0.193	0.357±0.203	0.702±0.227	0.747±0.220
POSS	0.487±0.291	0.487±0.290	0.488±0.293	0.491±0.292	0.522±0.368	0.498±0.306	0.556±0.392
POSSALPHA	0.488±0.290	0.486±0.290	0.488±0.294	0.498±0.290	0.526±0.357	0.498±0.306	0.552±0.396
POSSSTOP	0.477±0.287	0.477±0.287	0.471±0.282	0.467±0.283	0.518±0.389	0.486±0.297	0.540±0.388
POSSSTOPALPHA	0.469±0.283	0.470±0.284	0.471±0.288	0.465±0.280	0.517±0.407	0.478±0.297	0.525±0.376
TOK	0.793±0.182	0.788±0.182	0.793±0.182	0.796±0.173	0.632±0.337	0.716±0.221	0.711±0.272
TOKALPHA	0.768±0.197	0.768±0.193	0.757±0.202	0.773±0.191	0.271±0.175	0.705±0.224	0.721±0.261
TOKNER	0.789±0.185	0.785±0.179	0.788±0.182	0.789±0.183	0.609±0.344	0.708±0.225	0.703±0.271
TOKNERALPHA	0.768±0.194	0.771±0.194	0.763±0.200	0.776±0.189	0.628±0.325	0.696±0.230	0.701±0.270
TOKNERR	0.741±0.215	0.744±0.218	0.737±0.219	0.743±0.216	0.600±0.337	0.696±0.231	0.688±0.281
TOKNERRALPHA	0.734±0.215	0.735±0.220	0.735±0.228	0.730±0.217	0.624±0.320	0.683±0.235	0.681±0.280
TOKNERRSTOP	0.736±0.220	0.736±0.222	0.728±0.224	0.732±0.223	0.609±0.333	0.680±0.239	0.730±0.244
TOKNERRSTOPALPHA	0.728±0.225	0.731±0.222	0.727±0.237	0.723±0.229	0.623±0.319	0.675±0.243	0.721±0.249
TOKNERSTOP	0.785±0.186	0.791±0.186	0.790±0.186	0.790±0.181	0.635±0.324	0.703±0.228	0.732±0.230
TOKNERSTOPALPHA	0.773±0.197	0.771±0.191	0.762±0.208	0.774±0.193	0.646±0.309	0.691±0.233	0.737±0.236
TOKPOS	0.781±0.199	0.783±0.199	0.791±0.203	0.798±0.193	0.565±0.378	0.713±0.222	0.656±0.313
TOKPOSALPHA	0.775±0.203	0.775±0.206	0.778±0.214	0.784±0.208	0.576±0.191	0.699±0.223	0.653±0.304
TOKPOSS	0.766±0.203	0.768±0.200	0.767±0.201	0.783±0.195	0.549±0.372	0.715±0.224	0.648±0.268
TOKPOSSALPHA	0.765±0.195	0.761±0.196	0.763±0.196	0.767±0.192	0.378±0.277	0.709±0.228	0.662±0.258
TOKPOSSSTOP	0.763±0.184	0.765±0.184	0.767±0.182	0.773±0.174	0.563±0.307	0.704±0.223	0.703±0.317
TOKPOSSSTOPALPHA	0.774±0.192	0.773±0.193	0.774±0.189	0.771±0.186	0.671±0.302	0.694±0.223	0.722±0.318
TOKPOSSTOP	0.786±0.187	0.783±0.186	0.794±0.182	0.792±0.179	0.645±0.286	0.700±0.225	0.711±0.262
TOKPOSSTOPALPHA	-	-	-	-	-	-	-
TOKSTOP	0.793±0.180	0.790±0.182	0.784±0.188	0.794±0.180	0.644±0.327	0.708±0.222	0.758±0.216
TOKSTOPALPHA	0.775±0.188	0.776±0.190	0.766±0.199	0.776±0.192	0.342±0.194	0.700±0.228	0.745±0.217
Avg F1	0.705	0.705	0.696	0.704	0.508	0.660	0.655
Avg STDE	0.241	0.241	0.250	0.244	0.281	0.249	0.312

	AdaBoost	XGBoost	MLP	CNN1	CNN2	AvgF1	AvgSTDE
CHNK	0.649±0.298	0.667±0.289	0.724±0.310	0.657±0.256	0.666±0.278	0.677	0.284
CHNKALPHA	0.616±0.302	0.676±0.291	0.695±0.385	0.587±0.337	0.583±0.429	0.648	0.310
CHNKNER	0.630±0.299	0.673±0.292	0.722±0.318	0.654±0.196	0.642±0.287	0.673	0.280
CHNKNERALPHA	0.609±0.301	0.649±0.304	0.684±0.390	0.557±0.282	0.614±0.426	0.637	0.308
CHNKNERR	0.608±0.310	0.642±0.305	0.704±0.320	0.645±0.262	0.662±0.288	0.657	0.294
CHNKNERRALPHA	0.599±0.312	0.653±0.302	0.674±0.412	0.566±0.288	0.600±0.437	0.631	0.315
CHNKNERRSTOP	0.621±0.308	0.652±0.300	0.693±0.365	0.402±0.264	0.344±0.223	0.608	0.291
CHNKNERRSTOPALPHA	0.582±0.312	0.648±0.298	0.623±0.445	0.451±0.314	0.340±0.370	0.558	0.308
CHNKNERSTOP	0.652±0.291	0.679±0.285	0.720±0.296	0.501±0.200	0.298±0.195	0.635	0.268
CHNKNERSTOPALPHA	0.625±0.301	0.656±0.291	0.647±0.443	0.431±0.340	0.406±0.286	0.589	0.293
CHNKSTOP	0.648±0.290	0.679±0.279	0.715±0.278	0.386±0.257	0.342±0.196	0.630	0.268
CHNKSTOPALPHA	0.654±0.290	0.664±0.285	0.628±0.431	0.455±0.277	0.374±0.317	0.577	0.293
DEP	0.566±0.344	0.598±0.349	0.594±0.319	0.682±0.214	0.694±0.249	0.577	0.314
DEPALPHA	0.552±0.352	0.604±0.346	0.598±0.392	0.606±0.396	0.62±0.418	0.571	0.349
DEPNER	0.564±0.350	0.595±0.356	0.592±0.325	0.686±0.211	0.692±0.263	0.576	0.314
DEPNERALPHA	0.497±0.287	0.593±0.328	0.603±0.432	0.606±0.351	0.623±0.407	0.553	0.316
DEPNERR	0.562±0.350	0.596±0.357	0.595±0.334	0.670±0.227	0.695±0.299	0.574	0.319
DEPNERRALPHA	0.556±0.349	0.595±0.354	0.593±0.440	0.585±0.311	0.622±0.415	0.567	0.347
DEPNERRSTOP	0.572±0.350	0.600±0.354	-	0.726±0.225	0.702±0.249	0.579	0.320
DEPNERRSTOPALPHA	0.561±0.357	0.595±0.361	0.574±0.449	0.583±0.328	0.619±0.444	0.564	0.366
DEPNERSTOP	0.563±0.360	0.604±0.360	0.577±0.341	0.725±0.219	0.708±0.248	0.578	0.321
DEPNERSTOPALPHA	0.422±0.239	0.576±0.346	0.564±0.465	0.630±0.321	0.632±0.409	0.528	0.321
DEPSTOP	0.576±0.349	0.603±0.358	0.584±0.330	0.741±0.209	0.648±0.271	0.577	0.324
DEPSTOPALPHA	0.568±0.363	0.595±0.365	0.578±0.446	0.629±0.352	0.625±0.386	0.570	0.364
LEM	0.720±0.222	0.744±0.225	0.786±0.258	0.670±0.193	0.665±0.265	0.733	0.225
LEMALPHA	0.705±0.242	0.748±0.222	0.754±0.286	0.610±0.208	0.651±0.283	0.684	0.224
LEMNER	0.720±0.237	0.742±0.227	0.780±0.274	0.680±0.190	0.613±0.268	0.728	0.228
LEMNERALPHA	0.707±0.247	0.742±0.238	0.768±0.299	0.662±0.203	0.671±0.284	0.721	0.238
LEMNERR	0.683±0.260	0.724±0.246	0.749±0.286	0.658±0.203	0.663±0.231	0.702	0.245
LEMNERRALPHA	0.680±0.260	0.710±0.244	0.740±0.267	0.645±0.277	0.652±0.289	0.694	0.256
LEMNERRSTOP	0.690±0.258	0.720±0.242	0.741±0.343	0.371±0.306	0.364±0.214	0.653	0.255
LEMNERRSTOPALPHA	0.682±0.255	0.704±0.247	0.737±0.310	0.372±0.214	0.348±0.267	0.647	0.251
LEMNERSTOP	0.725±0.231	0.742±0.222	0.780±0.298	0.429±0.198	0.378±0.215	0.690	0.224
LEMNERSTOPALPHA	0.716±0.237	0.738±0.233	0.768±0.343	0.460±0.210	0.414±0.216	0.685	0.233
LEMPOS	0.727±0.264	0.741±0.244	0.783±0.251	0.665±0.194	0.640±0.264	0.715	0.243
LEMPOSALPHA	0.713±0.272	0.727±0.252	0.775±0.340	0.664±0.213	0.695±0.250	0.711	0.241
LEMPOSS	0.679±0.251	0.717±0.249	0.773±0.335	0.662±0.262	0.736±0.271	0.714	0.253
LEMPOSSALPHA	0.674±0.266	0.712±0.249	0.756±0.306	0.603±0.217	0.715±0.313	0.689	0.243
LEMPOSSSTOP	0.691±0.225	0.720±0.230	0.773±0.291	0.683±0.221	0.725±0.234	0.729	0.232
LEMPOSSSTOPALPHA	0.681±0.236	0.714±0.238	0.757±0.296	0.593±0.263	0.716±0.320	0.701	0.247
LEMPOSSTOP	0.721±0.232	0.735±0.231	0.783±0.353	0.715±0.260	0.707±0.270	0.738	0.238
LEMPOSSTOPALPHA	0.722±0.241	0.730±0.239	0.778±0.294	0.669±0.265	0.698±0.227	0.729	0.234
LEMSTOP	0.732±0.228	0.752±0.219	0.789±0.321	0.403±0.197	0.327±0.205	0.688	0.224
LEMSTOPALPHA	0.712±0.236	0.745±0.220	0.764±0.327	0.377±0.212	0.329±0.210	0.651	0.219
POSS	0.509±0.330	0.555±0.379	0.488±0.481	0.540±0.481	0.536±0.481	0.513	0.365
POSSALPHA	0.518±0.331	0.549±0.381	0.493±0.481	0.538±0.481	0.534±0.481	0.514	0.365
POSSSTOP	0.496±0.317	0.533±0.369	0.484±0.481	0.431±0.481	0.434±0.481	0.485	0.362
POSSSTOPALPHA	0.484±0.303	0.511±0.354	0.491±0.481	0.428±0.481	0.484±0.481	0.483	0.360
TOK	0.728±0.234	0.748±0.223	0.796±0.298	0.659±0.191	0.661±0.258	0.735	0.229
TOKALPHA	0.705±0.246	0.742±0.224	0.756±0.282	0.643±0.196	0.652±0.310	0.688	0.225
TOKNER	0.722±0.239	0.745±0.226	0.784±0.288	0.684±0.195	0.680±0.240	0.732	0.230
TOKNERALPHA	0.705±0.238	0.746±0.229	0.775±0.309	0.649±0.202	0.648±0.257	0.719	0.236
TOKNERR	0.671±0.263	0.719±0.246	0.749±0.291	0.655±0.200	0.631±0.275	0.698	0.249
TOKNERRALPHA	0.674±0.264	0.704±0.257	0.748±0.307	0.626±0.215	0.655±0.270	0.694	0.252
TOKNERRSTOP	0.678±0.260	0.710±0.253	0.751±0.303	0.406±0.212	0.317±0.225	0.651	0.247
TOKNERRSTOPALPHA	0.680±0.260	0.698±0.254	0.744±0.357	0.412±0.222	0.394±0.230	0.655	0.254
TOKNERSTOP	0.721±0.230	0.743±0.222	0.790±0.325	0.444±0.205	0.367±0.209	0.691	0.226
TOKNERSTOPALPHA	0.704±0.241	0.740±0.226	0.771±0.330	0.371±0.262	0.379±0.217	0.677	0.237
TOKPOS	0.720±0.270	0.739±0.255	0.787±0.344	0.626±0.201	0.705±0.227	0.722	0.250
TOKPOSALPHA	0.705±0.277	0.731±0.262	0.783±0.328	0.633±0.268	0.698±0.250	0.716	0.245
TOKPOSS	0.671±0.254	0.715±0.249	0.773±0.332	0.686±0.204	0.729±0.231	0.714	0.245
TOKPOSSALPHA	0.656±0.241	0.709±0.241	0.769±0.418	0.643±0.269	0.658±0.275	0.687	0.249
TOKPOSSSTOP	0.684±0.244	0.724±0.229	0.771±0.354	0.675±0.205	0.722±0.227	0.718	0.236
TOKPOSSSTOPALPHA	0.713±0.242	0.730±0.233	0.779±0.328	0.680±0.265	0.698±0.238	0.732	0.242
TOKPOSSTOP	0.733±0.230	0.739±0.232	0.789±0.338	0.706±0.200	0.691±0.226	0.739	0.228
TOKPOSSTOPALPHA	-	-	-	-	-	-	-
TOKSTOP	0.736±0.227	0.749±0.218	0.787±0.368	0.355±0.199	0.321±0.207	0.685	0.226
TOKSTOPALPHA	0.714±0.231	0.744±0.215	0.765±0.337	0.452±0.215	0.425±0.210	0.665	0.218
Avg F1	0.649	0.682	0.703	0.580	0.576		
Avg STDE	0.277	0.275	0.347	0.257	0.289		

Table A.2: Japanese dataset: F-scores and average standard error for classifier-preprocessing pairs (F1, stderr)

	LBFGS LR	NewtonLR	LinearSVM	SGD SVM	KNN	NaiveBayes	RandForest
CHNK	0.710±0.040	0.708±0.045	0.718±0.041	0.715±0.047	0.495±0.206	0.753±0.033	0.677±0.054
CHNKALPHA	0.714±0.028	0.719±0.029	0.709±0.031	0.702±0.040	0.510±0.179	0.740±0.027	0.684±0.043
CHNKNER	0.758±0.042	0.758±0.037	0.769±0.034	0.763±0.031	0.523±0.174	0.772±0.031	0.738±0.039
CHNKNERALPHA	0.751±0.031	0.755±0.030	0.757±0.031	0.751±0.034	0.555±0.159	0.767±0.029	0.725±0.040
CHNKNERR	0.752±0.030	0.751±0.030	0.756±0.032	0.745±0.038	0.517±0.151	0.757±0.030	0.727±0.044
CHNKNERRALPHA	0.733±0.046	0.736±0.040	0.744±0.035	0.737±0.037	0.529±0.163	0.745±0.023	0.712±0.034
CHNKNERRSTOP	0.745±0.036	0.744±0.040	0.753±0.026	0.737±0.038	0.524±0.165	0.739±0.040	0.729±0.042
CHNKNERRSTOPALPHA	0.731±0.042	0.730±0.048	0.741±0.036	0.723±0.037	0.539±0.160	0.730±0.026	0.718±0.053
CHNKNERSTOP	0.754±0.040	0.755±0.040	0.764±0.033	0.776±0.024	0.543±0.160	0.767±0.029	0.738±0.039
CHNKNERSTOPALPHA	0.758±0.037	0.755±0.032	0.760±0.032	0.761±0.027	0.528±0.171	0.750±0.022	0.721±0.045
CHNKSTOP	0.712±0.051	0.714±0.053	0.724±0.050	0.716±0.045	0.506±0.200	0.728±0.046	0.675±0.067
CHNKSTOPALPHA	0.695±0.049	0.701±0.050	0.714±0.048	0.700±0.049	0.518±0.191	0.721±0.034	0.666±0.071
DEP	0.678±0.069	0.681±0.070	0.685±0.066	0.680±0.055	0.495±0.188	0.682±0.061	0.608±0.124
DEPALPHA	0.669±0.074	0.668±0.074	0.681±0.067	0.674±0.065	0.464±0.194	0.674±0.061	0.593±0.137
DEPNER	0.727±0.053	0.727±0.045	0.742±0.035	0.746±0.033	0.502±0.171	0.724±0.040	0.708±0.059
DEPNERALPHA	0.730±0.046	0.731±0.048	0.741±0.032	0.748±0.036	0.427±0.248	0.718±0.043	0.707±0.065
DEPNERR	0.677±0.065	0.684±0.067	0.681±0.066	0.686±0.061	0.479±0.200	0.676±0.060	0.608±0.122
DEPNERRALPHA	0.667±0.074	0.671±0.077	0.675±0.066	0.674±0.070	0.461±0.187	0.670±0.055	0.585±0.137
DEPNERRSTOP	0.659±0.084	0.652±0.083	0.659±0.086	0.649±0.082	0.509±0.166	0.645±0.075	0.571±0.152
DEPNERRSTOPALPHA	0.617±0.108	0.617±0.109	0.619±0.109	0.618±0.102	0.493±0.167	0.627±0.099	0.534±0.180
DEPNERSTOP	0.717±0.055	0.716±0.053	0.728±0.047	0.723±0.034	0.544±0.130	0.701±0.041	0.693±0.062
DEPNERSTOPALPHA	0.715±0.050	0.711±0.050	0.722±0.039	0.713±0.041	0.448±0.237	0.731±0.032	0.695±0.063
DEPSTOP	0.655±0.083	0.658±0.079	0.658±0.082	0.653±0.076	0.507±0.166	0.647±0.074	0.575±0.148
DEPSTOPALPHA	0.627±0.109	0.623±0.105	0.622±0.100	0.622±0.094	0.499±0.169	0.628±0.093	0.537±0.168
LEM	0.803±0.020	0.799±0.022	0.817±0.017	0.802±0.026	0.592±0.115	0.823±0.020	0.771±0.032
LEMALPHA	0.800±0.021	0.808±0.032	0.824±0.024	0.807±0.022	0.587±0.115	0.815±0.022	0.758±0.030
LEMNER	0.806±0.028	0.801±0.032	0.818±0.023	0.816±0.022	0.633±0.110	0.823±0.029	0.790±0.022
LEMNERALPHA	0.809±0.028	0.807±0.022	0.827±0.020	0.823±0.014	0.612±0.102	0.827±0.029	0.797±0.026
LEMNERR	0.794±0.029	0.790±0.032	0.796±0.019	0.791±0.024	0.585±0.117	0.808±0.022	0.778±0.026
LEMNERRALPHA	0.797±0.027	0.796±0.021	0.805±0.027	0.793±0.021	0.585±0.114	0.811±0.024	0.781±0.032
LEMNERRSTOP	0.787±0.024	0.784±0.028	0.793±0.023	0.793±0.024	0.590±0.121	0.794±0.025	0.796±0.024
LEMNERRSTOPALPHA	0.790±0.024	0.791±0.027	0.798±0.026	0.793±0.025	0.587±0.130	0.796±0.027	0.788±0.031
LEMNERSTOP	0.799±0.030	0.798±0.025	0.819±0.028	0.823±0.021	0.632±0.099	0.831±0.026	0.802±0.028
LEMNERSTOPALPHA	0.805±0.025	0.808±0.029	0.830±0.024	0.826±0.020	0.643±0.100	0.834±0.014	0.802±0.019
LEMPOS	0.806±0.021	0.808±0.022	0.845±0.015	0.850±0.023	0.716±0.039	0.830±0.014	0.786±0.024
LEMPOSALPHA	0.807±0.026	0.803±0.035	0.838±0.017	0.846±0.019	0.704±0.030	0.828±0.026	0.778±0.037
LEMPOSS	0.815±0.023	0.817±0.017	0.851±0.021	0.842±0.017	0.711±0.178	0.855±0.025	0.785±0.033
LEMPOSSALPHA	0.825±0.020	0.823±0.030	0.848±0.018	0.837±0.019	0.734±0.035	0.847±0.020	0.770±0.039
LEMPOSSSTOP	0.816±0.025	0.818±0.028	0.850±0.018	0.846±0.017	0.529±0.038	0.851±0.018	0.774±0.031
LEMPOSSSTOPALPHA	0.819±0.036	0.817±0.025	0.841±0.025	0.829±0.016	0.722±0.037	0.844±0.022	0.779±0.034
LEMPOSSTOP	0.802±0.023	0.795±0.029	0.841±0.015	0.847±0.023	0.567±0.144	0.836±0.021	0.770±0.030
LEMPOSSTOPALPHA	0.791±0.017	0.789±0.024	0.830±0.026	0.836±0.019	0.700±0.028	0.837±0.022	0.768±0.035
LEMSTOP	0.799±0.028	0.801±0.026	0.813±0.019	0.802±0.024	0.614±0.132	0.812±0.032	0.785±0.030
LEMSTOPALPHA	0.803±0.031	0.804±0.030	0.813±0.025	0.807±0.031	0.619±0.132	0.821±0.025	0.776±0.032
POSS	0.645±0.038	0.646±0.030	0.634±0.039	0.627±0.046	0.558±0.051	0.620±0.035	0.645±0.041
POSSALPHA	0.646±0.036	0.650±0.039	0.635±0.051	0.619±0.067	0.557±0.055	0.618±0.032	0.644±0.038
POSSSTOP	0.641±0.034	0.643±0.016	0.634±0.030	0.637±0.050	0.553±0.083	0.626±0.038	0.638±0.022
POSSSTOPALPHA	0.615±0.034	0.616±0.034	0.603±0.063	0.610±0.052	0.506±0.125	0.614±0.045	0.610±0.035
TOK	0.801±0.024	0.797±0.022	0.813±0.023	0.806±0.021	0.592±0.144	0.817±0.022	0.766±0.029
TOKALPHA	0.801±0.020	0.799±0.025	0.807±0.019	0.804±0.020	0.584±0.135	0.817±0.019	0.761±0.031
TOKNER	0.814±0.028	0.812±0.027	0.830±0.018	0.828±0.020	0.718±0.118	0.835±0.018	0.763±0.031
TOKNERALPHA	0.798±0.025	0.800±0.021	0.816±0.015	0.820±0.026	0.610±0.117	0.825±0.018	0.790±0.028
TOKNERR	0.789±0.031	0.785±0.029	0.797±0.017	0.799±0.016	0.589±0.126	0.798±0.020	0.783±0.021
TOKNERRALPHA	0.791±0.022	0.792±0.027	0.802±0.024	0.792±0.024	0.567±0.126	0.797±0.019	0.778±0.029
TOKNERRSTOP	0.776±0.032	0.778±0.032	0.795±0.028	0.783±0.021	0.582±0.129	0.788±0.024	0.776±0.025
TOKNERRSTOPALPHA	0.785±0.024	0.787±0.035	0.796±0.022	0.795±0.025	0.571±0.127	0.783±0.032	0.778±0.029
TOKNERSTOP	0.803±0.026	0.805±0.031	0.825±0.022	0.824±0.026	0.608±0.114	0.822±0.014	0.790±0.031
TOKNERSTOPALPHA	0.794±0.030	0.794±0.031	0.819±0.035	0.820±0.015	0.616±0.116	0.820±0.019	0.797±0.031
TOKPOS	0.799±0.020	0.796±0.019	0.822±0.021	0.828±0.016	0.630±0.023	0.826±0.013	0.794±0.027
TOKPOSALPHA	0.808±0.015	0.807±0.022	0.844±0.025	0.852±0.022	0.706±0.041	0.828±0.018	0.777±0.040
TOKPOSS	0.815±0.025	0.812±0.014	0.843±0.022	0.835±0.023	0.695±0.191	0.849±0.018	0.775±0.025
TOKPOSSALPHA	0.788±0.032	0.789±0.028	0.829±0.022	0.833±0.014	0.716±0.022	0.823±0.015	0.752±0.045
TOKPOSSSTOP	0.823±0.025	0.814±0.024	0.834±0.027	0.828±0.014	0.713±0.036	0.848±0.022	0.767±0.029
TOKPOSSSTOPALPHA	0.817±0.023	0.822±0.019	0.837±0.021	0.835±0.013	0.518±0.041	0.846±0.020	0.764±0.036
TOKPOSSTOP	0.802±0.038	0.800±0.027	0.839±0.019	0.840±0.019	0.691±0.134	0.823±0.020	0.771±0.032
TOKPOSSTOPALPHA	-	0.794±0.029	0.834±0.021	0.844±0.023	0.560±0.033	0.832±0.015	0.766±0.034
TOKSTOP	0.803±0.027	0.801±0.027	0.815±0.028	0.803±0.025	0.584±0.133	0.809±0.023	0.778±0.041
TOKSTOPALPHA	0.800±0.026	0.800±0.032	0.811±0.024	0.813±0.019	0.591±0.132	0.821±0.019	0.779±0.027
Avg F1	0.758	0.758	0.773	0.769	0.579	0.772	0.729
Avg STDE	0.037	0.037	0.034	0.033	0.126	0.031	0.054

	AdaBoost	XGBoost	MLP	CNN1	CNN2	AvgF1	AvgSTDE
CHNK	0.592±0.122	0.632±0.091	0.761±0.029	0.771±0.026	0.635±0.026	0.681	0.063
CHNKALPHA	0.590±0.120	0.622±0.089	0.738±0.023	0.747±0.029	0.629±0.029	0.675	0.056
CHNKNER	0.696±0.057	0.714±0.041	0.796±0.019	0.789±0.039	-	0.734	0.049
CHNKNERALPHA	0.686±0.059	0.712±0.043	0.758±0.020	0.768±0.025	-	0.725	0.046
CHNKNERR	0.690±0.059	0.713±0.046	0.780±0.028	0.786±0.020	0.699±0.032	0.723	0.045
CHNKNERRALPHA	0.679±0.064	0.703±0.054	0.747±0.030	0.770±0.027	0.669±0.037	0.709	0.049
CHNKNERRSTOP	0.694±0.057	0.700±0.049	0.775±0.017	0.778±0.029	0.665±0.023	0.715	0.047
CHNKNERRSTOPALPHA	0.676±0.069	0.690±0.062	0.736±0.028	0.727±0.032	0.657±0.027	0.700	0.052
CHNKNERSTOP	0.696±0.053	0.709±0.047	0.784±0.027	0.791±0.025	-	0.737	0.047
CHNKNERSTOPALPHA	0.692±0.053	0.707±0.048	0.745±0.030	0.752±0.019	-	0.718	0.047
CHNKSTOP	0.580±0.123	0.629±0.092	0.753±0.023	0.746±0.030	0.688±0.021	0.681	0.067
CHNKSTOPALPHA	0.572±0.133	0.628±0.089	0.702±0.028	0.722±0.023	0.604±0.033	0.662	0.067
DEP	0.534±0.176	0.583±0.127	0.764±0.027	0.765±0.029	0.573±0.027	0.644	0.085
DEPALPHA	0.529±0.179	0.571±0.127	0.756±0.033	0.705±0.019	0.621±0.033	0.634	0.089
DEPNER	0.690±0.065	0.707±0.049	0.792±0.027	0.748±0.021	-	0.697	0.054
DEPNERALPHA	0.689±0.058	0.703±0.045	0.782±0.016	0.756±0.026	-	0.692	0.060
DEPNERR	0.520±0.184	0.578±0.128	0.791±0.035	0.773±0.026	0.590±0.037	0.645	0.087
DEPNERRALPHA	0.524±0.182	0.576±0.128	0.767±0.032	0.762±0.027	0.587±0.034	0.635	0.089
DEPNERRSTOP	0.514±0.190	0.549±0.152	0.783±0.018	0.761±0.025	0.479±0.034	0.619	0.095
DEPNERRSTOPALPHA	0.504±0.195	0.523±0.175	0.755±0.025	0.738±0.040	0.592±0.020	0.603	0.111
DEPNERSTOP	0.681±0.070	0.690±0.062	0.802±0.026	0.750±0.023	-	0.700	0.055
DEPNERSTOPALPHA	0.683±0.066	0.685±0.046	0.779±0.022	0.739±0.020	-	0.691	0.061
DEPSTOP	0.523±0.183	0.556±0.151	0.776±0.028	0.731±0.028	0.659±0.040	0.633	0.095
DEPSTOPALPHA	0.516±0.189	0.535±0.172	0.746±0.028	0.743±0.027	0.608±0.029	0.609	0.107
LEM	0.669±0.080	0.723±0.040	0.863±0.017	0.868±0.031	0.721±0.029	0.771	0.037
LEMALPHA	0.666±0.080	0.727±0.045	0.854±0.015	0.853±0.022	0.805±0.021	0.775	0.037
LEMNER	0.747±0.042	0.775±0.035	0.870±0.017	0.865±0.020	0.616±0.021	0.780	0.033
LEMNERALPHA	0.744±0.047	0.776±0.030	0.857±0.019	0.860±0.031	0.772±0.028	0.793	0.033
LEMNERR	0.744±0.044	0.763±0.037	0.843±0.018	0.811±0.021	0.654±0.022	0.763	0.034
LEMNERRALPHA	0.748±0.040	0.768±0.027	0.845±0.029	0.828±0.018	0.708±0.022	0.772	0.033
LEMNERRSTOP	0.740±0.037	0.756±0.039	0.839±0.019	0.836±0.016	0.807±0.016	0.776	0.033
LEMNERRSTOPALPHA	0.743±0.041	0.764±0.029	0.810±0.022	0.814±0.014	0.776±0.026	0.771	0.035
LEMNERSTOP	0.742±0.032	0.767±0.036	0.875±0.019	0.870±0.012	0.807±0.023	0.797	0.032
LEMNERSTOPALPHA	0.744±0.038	0.769±0.040	0.848±0.022	0.849±0.034	0.741±0.031	0.792	0.033
LEMPOS	0.742±0.051	0.768±0.026	0.868±0.021	0.870±0.025	0.619±0.024	0.792	0.025
LEMPOSALPHA	0.728±0.053	0.772±0.033	0.856±0.017	0.860±0.015	0.663±0.016	0.790	0.027
LEMPOSS	0.718±0.047	0.780±0.033	0.866±0.019	-	0.768±0.013	0.807	0.039
LEMPOSSALPHA	0.690±0.064	0.759±0.041	0.856±0.024	0.867±0.024	0.572±0.028	0.786	0.030
LEMPOSSSTOP	0.718±0.030	0.764±0.022	0.868±0.023	0.876±0.020	0.820±0.024	0.794	0.025
LEMPOSSSTOPALPHA	0.686±0.025	0.743±0.030	0.850±0.015	0.851±0.029	0.688±0.015	0.789	0.026
LEMPOSSTOP	0.730±0.034	0.769±0.018	0.868±0.022	0.877±0.027	0.808±0.025	0.793	0.034
LEMPOSSTOPALPHA	0.707±0.043	0.754±0.021	0.835±0.018	0.860±0.028	0.783±0.031	0.791	0.026
LEMSTOP	0.659±0.082	0.723±0.054	0.858±0.018	0.862±0.015	0.809±0.017	0.778	0.040
LEMSTOPALPHA	0.665±0.078	0.727±0.059	0.842±0.025	0.844±0.021	0.801±0.027	0.777	0.043
POSS	0.643±0.029	0.632±0.024	0.645±0.024	0.670±0.032	0.490±0.032	0.621	0.035
POSSALPHA	0.646±0.030	0.640±0.029	0.649±0.036	0.658±0.034	0.567±0.030	0.627	0.040
POSSSTOP	0.643±0.028	0.630±0.031	0.635±0.028	0.623±0.024	0.492±0.030	0.616	0.035
POSSSTOPALPHA	0.609±0.033	0.595±0.033	0.614±0.047	0.587±0.050	0.508±0.053	0.591	0.050
TOK	0.647±0.074	0.724±0.051	0.864±0.019	0.857±0.024	0.792±0.013	0.773	0.039
TOKALPHA	0.655±0.077	0.712±0.052	0.847±0.020	0.848±0.014	0.798±0.018	0.769	0.037
TOKNER	0.667±0.037	0.731±0.032	0.846±0.018	0.847±0.022	0.686±0.024	0.781	0.033
TOKNERALPHA	0.734±0.032	0.762±0.038	0.870±0.024	0.864±0.022	0.834±0.030	0.794	0.033
TOKNERR	0.729±0.037	0.756±0.040	0.838±0.028	0.844±0.016	0.793±0.022	0.775	0.034
TOKNERRALPHA	0.737±0.034	0.759±0.034	0.836±0.021	0.821±0.018	0.722±0.026	0.766	0.034
TOKNERRSTOP	0.727±0.039	0.756±0.034	0.835±0.020	0.843±0.014	0.709±0.027	0.762	0.035
TOKNERRSTOPALPHA	0.736±0.040	0.761±0.029	0.818±0.016	0.817±0.023	0.742±0.019	0.764	0.035
TOKNERSTOP	0.745±0.045	0.768±0.033	0.842±0.015	0.857±0.027	0.811±0.019	0.792	0.034
TOKNERSTOPALPHA	0.735±0.039	0.759±0.033	0.878±0.028	0.875±0.027	0.849±0.041	0.796	0.037
TOKPOS	0.730±0.042	0.763±0.036	0.846±0.025	0.847±0.022	0.773±0.031	0.788	0.025
TOKPOSALPHA	0.724±0.056	0.770±0.030	0.870±0.015	0.865±0.019	0.674±0.019	0.794	0.027
TOKPOSS	0.719±0.057	0.756±0.031	0.876±0.031	-	0.683±0.021	0.795	0.042
TOKPOSSALPHA	0.715±0.068	0.748±0.048	0.864±0.019	0.865±0.020	0.646±0.014	0.781	0.029
TOKPOSSSTOP	0.701±0.025	0.760±0.021	0.855±0.022	0.821±0.021	0.643±0.022	0.784	0.024
TOKPOSSSTOPALPHA	0.703±0.029	0.764±0.021	0.876±0.031	0.883±0.020	0.759±0.019	0.785	0.024
TOKPOSSTOP	0.715±0.030	0.763±0.025	0.839±0.014	0.857±0.022	0.615±0.014	0.780	0.033
TOKPOSSTOPALPHA	0.720±0.044	0.762±0.032	-	-	-	0.775	0.029
TOKSTOP	0.646±0.078	0.717±0.055	0.857±0.013	0.861±0.013	0.782±0.023	0.771	0.041
TOKSTOPALPHA	0.650±0.080	0.717±0.058	0.827±0.016	0.818±0.021	0.710±0.050	0.761	0.042
Avg F1	0.673	0.707	0.809	0.806	0.688		
Avg STDE	0.049	0.023	0.026	0.031	0.036		

Bibliography

[1] J. W. Patchin and S. Hinduja. Bullies move beyond the schoolyard: a preliminary look at cyberbullying. *Youth Violence and Juvenile Justice*, 4(2):148–169, 2006. DOI: 10.1177/1541204006286288. eprint: https://doi.org/10.1177/1541204006286288. URL: https://doi.org/10.1177/1541204006286288.

[2] G. Bull. The always-connected generation. *Learning and Leading with Technology*, 38:28–29, 2010.

[3] S. Hinduja and J. Patchin. Bullying, cyberbullying, and suicide. *Archives of suicide research : official journal of the International Academy for Suicide Research*, 14:206–21, July 2010. DOI: 10.1080/13811118.2010.494133.

[4] G. Sarna and M. Bhatia. Content based approach to find the credibility of user in social networks: an application of cyberbullying. *Int. J. Mach. Learn. and Cyber*, 8:677–689, 2015. DOI: 10.1007/s13042-015-0463-1.

[5] M. Ptaszynski and F. Masui. *Automatic Cyberbullying Detection: Emerging Research and Opportunities*. IGI Global, 2018.

[6] B. Vidgen and L. Derczynski. Directions in abusive language training data, a systematic review: garbage in, garbage out. *PLOS ONE*, 15(12):1–32, Dec. 2021. DOI: 10.1371/journal.pone.0243300. URL: https://doi.org/10.1371/journal.pone.0243300.

[7] A. Kontostathis, K. Reynolds, A. Garron, and L. Edwards. Detecting cyberbullying: query terms and techniques. In *Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13, pages 195–204, Paris, France. Association for Computing Machinery, 2013. ISBN: 9781450318891. DOI: 10.1145/2464464.2464499. URL: https://doi.org/10.1145/2464464.2464499.

[8] K. Dinakar, B. Jones, C. Havasi, H. Lieberman, and R. Picard. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Transactions on Interactive Intelligent Systems*, 2, Sept. 2012. DOI: 10.1145/2362394.2362400.

[9] M. Ptaszynski, P. Dybala, T. Matsuba, F. Masui, R. Rzepka, and K. Araki. Machine learning and affect analysis against cyber-bullying. In *Linguistic And Cognitive Approaches To Dialog Agents Symposium*, Mar. 2010.

[10] M. Ptaszynski, P. Dybala, T. Matsuba, F. Masui, R. Rzepka, K. Araki, and Y. Momouchi. In the service of online order: tackling cyber-bullying with machine learning and affect analysis. *International Journal of Computational Linguistics Research*, 1(3):135–154, 2010.

[11] M. Ptaszynski, F. Masui, Y. Kimura, R. Rzepka, and K. Araki. Extracting patterns of harmful expressions for cyberbullying detection. In *7th Language and Technology Conference (LTC'15), The First Workshop on Processing Emotions*, Nov. 2015.

[12] H. Mubarak, K. Darwish, and W. Magdy. Abusive language detection on arabic social media. In *Proceedings of the first workshop on abusive language online*, pages 52–56, 2017.

[13] A. Sahni and N. Raja. Analyzation and detection of cyberbullying: a twitter based indian case study. In *International Conference on Recent Developments in Science, Engineering and Technology*, pages 484–497. Springer, 2017.

[14] MEXT. 'netto-jō no ijime' ni kansuru taiō manyuaru jirei shū (gakkō, kyōin muke) ["bullying on the net" manual for handling and collection of cases (for schools and teachers)] (in japanese). *Ministry of Education, Culture, Sports, Science and Technology (MEXT)*, 2008. URL: `http://www.mext.go.jp/b_menu/houdou/20/11/08111701/001.pdf`.

[15] M. Ptaszynski, F. Masui, T. Nitta, S. Hatakeyama, Y. Kimura, R. Rzepka, and K. Araki. Sustainable cyberbullying detection with category-maximized relevance of harmful phrases and double-filtered automatic optimization. *International Journal of Child-Computer Interaction*, 8, Aug. 2016. DOI: `10.1016/j.ijcci.2016.07.002`.

[16] M. Ptaszynski, J. K. K. Eronen, and F. Masui. Learning deep on cyberbullying is always better than brute force. In *LaCATODA 2017 CEUR Workshop Proceedings*, pages 3–10, 2017. URL: `http://ceur-ws.org/Vol-1926/paper1.pdf`.

[17] A. A. Rahane and A. Subramanian. Measures of complexity for large scale image datasets. *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Feb. 2020. DOI: `10.1109/icaiic48513.2020.9065274`. URL: `http://dx.doi.org/10.1109/ICAIIC48513.2020.9065274`.

[18] A. Cutler. Lexical complexity and sentence processing. In G. B. Flores d'Arcais and R. J. Jarvella, editors, *The Process of Language Understanding*, chapter 2, pages 43–79. Chichester: Wiley, 1983.

[19] K. Rayner and S. A. Duffy. Lexical complexity and fixation times in reading: effects of word frequency, verb complexity, and lexical ambiguity. *Memory & cognition*, 14(3):191–201, 1986.

[20] T. Fellner and M. Apple. Developing writing fluency and lexical complexity with blogs. *The jalt call Journal*, 2(1):15–26, 2006.

[21] M. D. Johnson, L. Mercado, and A. Acevedo. The effect of planning subprocesses on l2 writing fluency, grammatical complexity, and lexical complexity. *Journal of Second Language Writing*, 21(3):264–282, 2012.

[22] J. Ure. Lexical density and register differentiation. *Applications of Linguistics*:443–452, 1971.

[23] F. Ferreira. Effects of length and syntactic complexity on initiation times for prepared utterances. *Journal of Memory and Language*, 30(2):210–233, 1991.

[24] S. M. Sotillo. Discourse functions and syntactic complexity in synchronous and asyncronous communication. *Language learning & technology*, 4(1):77–110, 2000.

[25] X. Lu. Automatic analysis of syntactic complexity in second language writing. *International journal of corpus linguistics*, 15(4):474–496, 2010.

[26] S. Hinduja and J. W. Patchin. Cyberbullying research center. `https://cyberbullying.org/`, 2021.

[27] T. Ranasinghe and M. Zampieri. Multilingual offensive language identification with cross-lingual embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 5838–5844, Nov. 2020.

[28] T. Ranasinghe and M. Zampieri. Multilingual offensive language identification for low-resource languages, 2021. arXiv: `2105.05996 [cs.CL]`.

[29] I. Bigoulaeva, V. Hangya, and A. Fraser. Cross-lingual transfer learning for hate speech detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 15–25, Kyiv. Association for Computational Linguistics, Apr. 2021. URL: `https://aclanthology.org/2021.ltedi-1.3`.

[30] S. Gaikwad, T. Ranasinghe, M. Zampieri, and C. M. Homan. Cross-lingual offensive language identification for low resource languages: the case of marathi, 2021. arXiv: `2109.03552 [cs.CL]`.

[31] R. Cotterell and G. Heigold. Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 748–759, Copenhagen, Denmark. Association for Computational Linguistics, Sept. 2017. DOI: 10.18653/v1/D17-1078. URL: https://aclanthology.org/D17-1078.

[32] C. Gooskens, V. J. van Heuven, J. Golubović, A. Schüppert, F. Swarte, and S. Voigt. Mutual intelligibility between closely related languages in europe. *International Journal of Multilingualism*, 15(2):169–193, 2018. DOI: 10.1080/14790718.2017.1350185. eprint: https://doi.org/10.1080/14790718.2017.1350185. URL: https://doi.org/10.1080/14790718.2017.1350185.

[33] S. Ruder, I. Vulić, and A. Søgaard. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631, 2019.

[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: pre-training of deep bidirectional transformers for language understanding, 2018. arXiv: 1810.04805 [cs.CL].

[35] J. Pyżalski. From cyberbullying to electronic aggression: typology of the phenomenon. *Emotional and Behavioural Difficulties*, 17(3-4):305–317, 2012. DOI: 10.1080/13632752.2012.704319. eprint: https://www.tandfonline.com/doi/pdf/10.1080/13632752.2012.704319. URL: https://www.tandfonline.com/doi/abs/10.1080/13632752.2012.704319.

[36] S. O. Sood, E. F. Churchill, and J. Antin. Automatic identification of personal insults on social news sites. *J. Am. Soc. Inf. Sci. Technol.*, 63(2):270–285, 2012. ISSN: 1532-2882. DOI: 10.1002/asi.21690. URL: https://doi.org/10.1002/asi.21690.

[37] A. Cano Basave, K. Liu, and J. Zhao. A weakly supervised bayesian model for violence detection in social media. In *6th International Joint Conference on Natural Language Processing (IJCNLP)*, Oct. 2013.

[38] T. Nitta, F. Masui, M. Ptaszynski, Y. Kimura, R. Rzepka, and K. Araki. Detecting cyberbullying entries on informal school websites based on category relevance maximization. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 579–586, Nagoya, Japan. Asian Federation of Natural Language Processing, 2013. URL: https://www.aclweb.org/anthology/I13-1066.

[39] P. D. Turney and M. L. Littman. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. *CoRR*, cs.LG/0212012, 2002. URL: http://arxiv.org/abs/cs/0212012.

[40] S. Hatakeyama, F. Masui, M. Ptaszynski, and K. Yamamoto. Statistical analysis of automatic seed word acquisition to improve harmful expression extraction in cyberbullying detection. *International Journal of Engineering and Technology Innovation*, 6(2):165–172, 2016.

[41] M. Ptaszynski, F. Masui, Y. Kimura, R. Rzepka, and K. Araki. Brute force works best against bullying. In *Proceedings of the 2015 International Conference on Constraints and Preferences for Configuration and Recommendation and Intelligent Techniques for Web Personalization - Volume 1440*, CPCR+ITWP'15, pages 28–29, Buenos Aires, Argentina. CEUR-WS.org, 2015.

[42] S. Agrawal and A. Awekar. Deep learning for detecting cyberbullying across multiple social media platforms. *CoRR*, abs/1801.06482, 2018. eprint: 1801.06482. URL: http://arxiv.org/abs/1801.06482.

[43] M. Mozafari, R. Farahbakhsh, and N. Crespi. A bert-based transfer learning approach for hate speech detection in online social media. In H. Cherifi, S. Gaito, J. F. Mendes, E. Moro, and L. M. Rocha, editors, *Complex Networks and Their Applications VIII*, pages 928–940, Cham. Springer International Publishing, 2020.

[44] M. Dadvar and K. Eckert. Cyberbullying detection in social networks using deep learning based models. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 245–255. Springer, 2020.

[45] J. Yadav, D. Kumar, and D. Chauhan. Cyberbullying detection using pretrained bert model. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 1096–1100, 2020. DOI: 10.1109/ICESC48915.2020.9155700.

[46] V. Balakrishnan, S. Khan, and H. R. Arabnia. Improving cyberbullying detection using twitter users' psychological features and machine learning. *Computers & Security*, 90, 2020. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2019.101710. URL: https://www.sciencedirect.com/science/article/pii/S0167404819302470. Article 101710.

[47] H. Rosa, N. Pereira, R. Ribeiro, P. C. Ferreira, J. P. Carvalho, S. Oliveira, L. Coheur, P. Paulino, A. V. Simão, and I. Trancoso. Automatic cyberbullying detection: a systematic review. *Computers in Human Behavior*, 93:333–345, 2019. ISSN: 0747-5632. DOI: https://doi.org/10.1016/j.chb.2018.12.021. URL: https://www.sciencedirect.com/science/article/pii/S0747563218306071.

[48] E. W. Pamungkas, V. Basile, and V. Patti. Towards multidomain and multilingual abusive language detection: a survey. *Personal and Ubiquitous Computing*:1–27, 2021.

[49] A. Basavanhally, S. Doyle, and A. Madabhushi. Predicting classifier performance with a small training set: applications to computer-aided diagnosis and prognosis. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 229–232, 2010.

[50] A. Basavanhally, S. Viswanath, and A. Madabhushi. Predicting classifier performance with limited training data: applications to computer-aided diagnosis in breast and prostate cancer. *PLOS ONE*, 10(5):1–18, May 2015. DOI: 10.1371/journal.pone.0117900. URL: https://doi.org/10.1371/journal.pone.0117900.

[51] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1):3133–3181, 2014.

[52] M. Johnson, P. Anderson, M. Dras, and M. Steedman. Predicting accuracy on large datasets from smaller pilot data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 450–455, Melbourne, Australia. Association for Computational Linguistics, July 2018. DOI: 10.18653/v1/P18-2072. URL: https://www.aclweb.org/anthology/P18-2072.

[53] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016. arXiv: 1607.01759. URL: http://arxiv.org/abs/1607.01759.

[54] J. Gama and P. Brazdil. Characterization of classification algorithms. In C. Pinto-Ferreira and N. J. Mamede, editors, *Progress in Artificial Intelligence*, pages 189–200, Berlin, Heidelberg. Springer Berlin Heidelberg, 1995. ISBN: 978-3-540-45595-0.

[55] R. D. King, C. Feng, and A. Sutherland. Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, 9(3):289–333, 1995. DOI: 10.1080/08839519508945477. eprint: https://doi.org/10.1080/08839519508945477. URL: https://doi.org/10.1080/08839519508945477.

[56] H. Bensusan and A. Kalousis. Estimating the predictive accuracy of a classifier. In L. De Raedt and P. Flach, editors, *Machine Learning: ECML 2001*, pages 25–36, Berlin, Heidelberg. Springer Berlin Heidelberg, 2001. ISBN: 978-3-540-44795-5.

[57] M. Blachnik. Instance selection for classifier performance estimation in meta learning. *Entropy*, 19:583, Nov. 2017. DOI: 10.3390/e19110583.

[58] P. P. Roy, J. T. Leonard, and K. Roy. Exploring the impact of size of training sets for the development of predictive qsar models. *Chemometrics and Intelligent Laboratory Systems*, 90(1):31–42, 2008. ISSN: 0169-7439. DOI: https://doi.org/10.1016/j.chemolab.2007.07.004. URL: http://www.sciencedirect.com/science/article/pii/S0169743907001529.

[59] C. Catal and B. Diri. Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem. *Information Sciences*, 179(8):1040–1058, 2009. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2008.12.001. URL: http://www.sciencedirect.com/science/article/pii/S0020025508005173.

[60] J. G. A. Barbedo. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, 153:46–53, 2018. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2018.08.013. URL: http://www.sciencedirect.com/science/article/pii/S0168169918304617.

[61] W. Chen, Y. Su, Y. Shen, Z. Chen, X. Yan, and W. Wang. How large a vocabulary does text classification need? a variational approach to vocabulary selection. *arXiv preprint arXiv:1902.10339*, 2019.

[62] W. Kusters. *Linguistic complexity*. Netherlands Graduate School of Linguistics, 2003.

[63] H. Akaike. A new look at statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

[64] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[65] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao. Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *arXiv preprint arXiv:1510.06168*, 2015.

[66] H. Demir and A. Özgür. Improving named entity recognition for morphologically rich languages using word embeddings. In *2014 13th International Conference on Machine Learning and Applications*, pages 117–122. IEEE, 2014.

[67] T. Schuster, O. Ram, R. Barzilay, and A. Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. *arXiv preprint arXiv:1902.09492*, 2019.

[68] O. Levy and Y. Goldberg. Dependency-based word embeddings. In *ACL*, 2014.

[69] A. Komninos and S. Manandhar. Dependency based embeddings for sentence classification tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1490–1500, San Diego, California. Association for Computational Linguistics, June 2016. DOI: 10.18653/v1/N16-1175. URL: https://www.aclweb.org/anthology/N16-1175.

[70] R. Cotterell and H. Schütze. Morphological word embeddings. *CoRR*, abs/1907.02423, 2019. arXiv: 1907.02423. URL: http://arxiv.org/abs/1907.02423.

[71] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[72] S. MacAvaney and A. Zeldes. A deeper look into dependency-based word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 40–45, New Orleans, Louisiana, USA. Association for Computational Linguistics, June 2018. DOI: 10.18653/v1/N18-4006. URL: https://www.aclweb.org/anthology/N18-4006.

[73] A. Mnih and G. Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 641–648, Corvalis, Oregon, USA. Association for Computing Machinery, 2007. ISBN: 9781595937933. DOI: 10.1145/1273496.1273577. URL: https://doi.org/10.1145/1273496.1273577.

[74] H. Ringbom. *Cross-linguistic Similarity in Foreign Language Learning*. Multilingual Matters, 2006. ISBN: 9781853599361. DOI: doi:10.21832/9781853599361. URL: https://doi.org/10.21832/9781853599361.

[75] R. Bley-Vroman. The evolving context of the fundamental difference hypothesis. *Studies in Second Language Acquisition*, 31(2):175–198, 2009. DOI: 10.1017/S0272263109090275.

[76] R. Cotterell, S. J. Mielke, J. Eisner, and B. Roark. Are all languages equally hard to language-model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 536–541, New Orleans, Louisiana. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/N18-2085. URL: https://aclanthology.org/N18-2085.

[77] V. Beaufils and J. Tomin. Stochastic approach to worldwide language classification: the signals and the noise towards long-range exploration, Oct. 2020. DOI: 10.31235/osf.io/5swba. URL: osf.io/preprints/socarxiv/5swba.

[78] L. Kovacevic, V. Bradic, G. de Melo, S. Zdravkovic, and O. Ryzhova. Ezglot. https://www.ezglot.com/, 2021.

[79] N. Aggarwal, T. Wunner, M. Arčan, P. Buitelaar, and S. O'Riain. A similarity measure based on semantic, terminological and linguistic information. In *Proceedings of the 6th International Workshop on Ontology Matching*, Jan. 2011.

[80] P. Achananuparp, X. Hu, and X. Shen. The evaluation of sentence similarity measures. In *Proceedings of the International Conference on Data Warehousing and Knowledge Discovery*, volume 5182, pages 305–316, Sept. 2008. ISBN: 978-3-540-85835-5. DOI: 10.1007/978-3-540-85836-2_29.

[81] A. Islam and D. Inkpen. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Trans. Knowl. Discov. Data*, 2(2), 2008. ISSN: 1556-4681. DOI: 10.1145/1376815.1376819. URL: https://doi.org/10.1145/1376815.1376819.

[82] P. Littell, D. R. Mortensen, K. Lin, K. Kairis, C. Turner, and L. Levin. URIEL and lang2vec: representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics, Apr. 2017. URL: https://aclanthology.org/E17-2002.

[83] M. S. Dryer and M. Haspelmath, editors. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig, 2013. URL: https://wals.info/.

[84] S. Moran, D. McCloy, and R. Wright, editors. *PHOIBLE Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig, 2014. URL: http://phoible.org/.

[85] D. M. Eberhard, G. F. Simons, and C. D. Fennig, editors. *Ethnologue: Languages of the World*. SIL International, Dallas, TX, USA, twenty-fifth edition, 2022. URL: http://www.ethnologue.com.

[86] H. Hammarström, R. Forkel, M. Haspelmath, and S. Bank. Glottolog/glottolog: glottolog database 4.5, version v4.5, Dec. 2021. DOI: 10.5281/zenodo.5772642. URL: https://doi.org/10.5281/zenodo.5772642.

[87] B. Szmrecsanyi. Geography is overrated. *Dialectological and folk dialectological concepts of space*:215–231, 2012.

[88] E. P. Stabler and E. L. Keenan. Structural similarity within and among languages. *Theoretical Computer Science*, 293(2):345–363, 2003. ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(01)00351-6. URL: https://www.sciencedirect.com/science/article/pii/S0304397501003516. Algebraic Methods in Language Processing.

[89] J. Hirschberg and C. D. Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015. DOI: 10.1126/science.aaa8685. eprint: https://www.science.org/doi/pdf/10.1126/science.aaa8685. URL: https://www.science.org/doi/abs/10.1126/science.aaa8685.

[90] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson. Xtreme: a massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR, 2020.

[91] Y. Liang, N. Duan, Y. Gong, N. Wu, F. Guo, W. Qi, M. Gong, L. Shou, D. Jiang, G. Cao, et al. Xglue: a new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, 2020.

[92] S. Ruder, N. Constant, J. Botha, A. Siddhant, O. Firat, J. Fu, P. Liu, J. Hu, D. Garrette, G. Neubig, and M. Johnson. XTREME-R: towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics, Nov. 2021. DOI: 10.18653/v1/2021.emnlp-main.802. URL: https://aclanthology.org/2021.emnlp-main.802.

[93] M. Pikuliak, M. Šimko, and M. Bieliková. Cross-lingual learning for text processing: a survey. *Expert Systems with Applications*, 165:113765, 2021. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113765. URL: https://www.sciencedirect.com/science/article/pii/S0957417420305893.

[94] I. Turc, K. Lee, J. Eisenstein, M.-W. Chang, and K. Toutanova. Revisiting the primacy of english in zero-shot cross-lingual transfer, 2021. DOI: 10.48550/ARXIV.2106.16171. URL: https://arxiv.org/abs/2106.16171.

[95] L. Duong, T. Cohn, S. Bird, and P. Cook. Cross-lingual transfer for unsupervised dependency parsing without parallel data. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 113–122, 2015.

[96] X. Chen, A. H. Awadallah, H. Hassan, W. Wang, and C. Cardie. Multi-source cross-lingual model transfer: learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3098–3112, Florence, Italy. Association for Computational Linguistics, July 2019. DOI: 10.18653/v1/P19-1299. URL: https://aclanthology.org/P19-1299.

[97] Q. Do and J. Gaspers. Cross-lingual transfer learning with data selection for large-scale spoken language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1455–1460, Hong Kong, China. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1153. URL: https://aclanthology.org/D19-1153.

[98] N. van der Heijden, H. Yannakoudakis, P. Mishra, and E. Shutova. Multilingual and cross-lingual document classification: a meta-learning approach. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1966–1976, Online. Association for Computational Linguistics, Apr. 2021. URL: https://aclanthology.org/2021.eacl-main.168.

[99] F. Nooralahzadeh, G. Bekoulis, J. Bjerva, and I. Augenstein. Zero-shot cross-lingual transfer with meta learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4547–4562, Online. Association for Computational Linguistics, Nov. 2020. DOI: 10.18653/v1/2020.emnlp-main.368. URL: https://aclanthology.org/2020.emnlp-main.368.

[100] Y.-H. Lin, C.-Y. Chen, J. Lee, Z. Li, Y. Zhang, M. Xia, S. Rijhwani, J. He, Z. Zhang, X. Ma, A. Anastasopoulos, P. Littell, and G. Neubig. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics, July 2019. DOI: 10.18653/v1/P19-1301. URL: https://aclanthology.org/P19-1301.

[101] A. Lauscher, V. Ravishankar, I. Vulić, and G. Glavaš. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics, Nov. 2020. DOI: 10.18653/v1/2020.emnlp-main.363. URL: https://aclanthology.org/2020.emnlp-main.363.

[102] A. Martinez-Garcia, T. Badia, and J. Barnes. Evaluating morphological typology in zero-shot cross-lingual transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3136–3153, Online. Association for Computational Linguistics, Aug. 2021. DOI: 10.18653/v1/2021.acl-long.244. URL: https://aclanthology.org/2021.acl-long.244.

[103] K. Reynolds, A. Edwards, and L. Edwards. Using machine learning to detect cyberbullying. *Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011*, 2, Dec. 2011. DOI: 10.1109/ICMLA.2011.152.

[104] M. Ptaszynski, G. Leliwa, M. Piech, and A. Smywiński-Pohl. Cyberbullying detection – technical report 2/2018, department of computer science agh, university of science and technology, 2018. arXiv: 1808.00926 [cs.CL].

[105] S. Hinduja and J. W. Patchin. *Bullying beyond the schoolyard: Preventing and responding to cyberbullying*. Corwin Press, 2014.

[106] M. Ptaszynski), A. Pieciukiewicz, and P. Dybała. Results of the poleval 2019 shared task 6: first dataset and open shared task for automatic cyberbullying detection in polish twitter. In *PolEval 2019 Workshop*, pages 89–110, 2019.

[107] M. Ptaszynski, P. Lempa, F. Masui, Y. Kimura, R. Rzepka, K. Araki, M. Wroczynski, and G. Leliwa. Brute - force sentence pattern extortion from harmful messages for cyberbullying detection. *Journal of the Association for Information Systems*, 20:8, 2019. DOI: 10.17705/1jais.00562. URL: https://aisel.aisnet.org/jais/vol20/iss8/4.

[108] M. Honnibal and I. Montani. spaCy 2: natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[109] F. Chollet et al. Keras. https://keras.io, 2015.

[110] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sept. 1995. ISSN: 1573-0565. DOI: 10.1007/BF00994018. URL: https://doi.org/10.1007/BF00994018.

[111] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[112] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2013.

[113] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1504. URL: http://www.sciencedirect.com/science/article/pii/S002200009791504X.

[114] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016. arXiv: 1603.02754. URL: http://arxiv.org/abs/1603.02754.

[115] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. arXiv: 1207.0580. URL: http://arxiv.org/abs/1207.0580.

[116] R. Collobert and J. Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167, Jan. 2008. DOI: 10.1145/1390156.1390177.

[117] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics, Oct. 2014. DOI: 10.3115/v1/D14-1181. URL: https://www.aclweb.org/anthology/D14-1181.

[118] D. Scherer, A. Müller, and S. Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *ICANN 2010 Proceedings, Part III*, pages 92–101, Jan. 2010. DOI: 10.1007/978-3-642-15825-4_10.

[119] Y. LeCun, L. Bottou, G. Orr., and K.-R. Muller. *Efficient backprop*. In *Neural Networks: Tricks of the Trade: Second Edition*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pages 9–48. ISBN: 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_3. URL: https://doi.org/10.1007/978-3-642-35289-8_3.

[120] N. Chawla, K. Bowyer, L. Hall, and P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002. ISSN: 1076-9757.

[121] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: a robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[122] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. English. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA, May 2010.

[123] M. Wiegand, M. Siegel, and J. Ruppenhofer. Overview of the germeval 2018 shared task on the identification of offensive language. In *GermEval 2018 Shared Task on the Identification of Offensive Language*, Sept. 2018.

[124] G. I. Sigurbergsson and L. Derczynski. Offensive language and hate speech detection for Danish. English. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3498–3508, Marseille, France. European Language Resources Association, May 2020. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.430.

[125] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics, June 2019. DOI: 10.18653/v1/N19-1144. URL: https://aclanthology.org/N19-1144.

[126] S. Smetanin. Toxic comments detection in russian. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2020"*, June 2020. DOI: 10.28995/NNNN-NNNN-2020-19-1-11.

[127] M. Arata. *Study on Change of Detection Accuracy Over Time in Cyberbullying Detection*. Master's thesis, Kitami Institute of Technology, Department of Computer Science, 2019.

[128] I. Takenaka, M. Ochiai, and Y. Matsui. The situation of occupational stress and related factors of harmful information countermeasure workers (in japanese). *Social psychology research (Japanese Society of Social Psychology)*, 33(3):135–148, 2018. DOI: 10.14966/jssp.1609.

[129] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang. On the opportunities and risks of foundation models, 2021. arXiv: 2108.07258 [cs.LG].

[130] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[131] K. K, Z. Wang, S. Mayhew, and D. Roth. Cross-lingual ability of multilingual bert: an empirical study. In *International Conference on Learning Representations*, 2020. URL: https://openreview.net/forum?id=HJeT3yrtDr.

[132] S. Wu and M. Dredze. Beto, bentz, becas: the surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics, Nov. 2019. DOI: 10.18653/v1/D19-1077. URL: https://aclanthology.org/D19-1077.

[133] T. Pires, E. Schlinger, and D. Garrette. How multilingual is multilingual bert? In *ACL*, 2019.

[134] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747.

[135] A. Conneau and G. Lample. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32:7059–7069, 2019.

[136] C. Brown, E. Holman, and S. Wichmann. Sound correspondences in the world's languages. *Language*, 89:4–29, Mar. 2013. DOI: 10.2307/23357720.

[137] C. Gooskens. The contribution of linguistic factors to the intelligibility of closely related languages. *Journal of Multilingual and multicultural development*, 28(6):445–467, 2007.

[138] G. Glavaš, M. Karan, and I. Vulić. XHate-999: analyzing and detecting abusive language across domains and languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6350–6365, Barcelona, Spain (Online). International Committee on Computational Linguistics, Dec. 2020. DOI: 10.18653/v1/2020.coling-main.559. URL: https://aclanthology.org/2020.coling-main.559.

[139] Q. Chen, W. Li, Y. Lei, X. Liu, and Y. He. Learning to adapt credible knowledge in cross-lingual sentiment analysis. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 419–429, 2015.

[140] R. Ghasemi, S. A. Ashrafi Asli, and S. Momtazi. Deep persian sentiment analysis: cross-lingual training for low-resource languages. *Journal of Information Science*:0165551520962781, 2020.

[141] L. Duong, T. Cohn, S. Bird, and P. Cook. Low resource dependency parsing: cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers)*, pages 845–850, 2015.

[142] Y. Wang, W. Che, J. Guo, Y. Liu, and T. Liu. Cross-lingual bert transformation for zero-shot dependency parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5721–5727, 2019.

[143] M. S. Bari, S. Joty, and P. Jwalapuram. Zero-resource cross-lingual named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7415–7423, Apr. 2020. DOI: 10.1609/aaai.v34i05.6237. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6237.

[144] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[145] K. Chakraborty, S. Bhattacharyya, and R. Bag. A survey of sentiment analysis from social media data. *IEEE Transactions on Computational Social Systems*, 7(2):450–464, 2020. DOI: 10.1109/TCSS.2019.2956957.

[146] A. Yadav and D. K. Vishwakarma. Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, 53(6):4335–4385, Aug. 2020. ISSN: 0269-2821. DOI: 10.1007/s10462-019-09794-5. URL: https://doi.org/10.1007/s10462-019-09794-5.

[147] H. Xu, B. Liu, L. Shu, and P. Yu. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics, June 2019. DOI: 10.18653/v1/N19-1242. URL: https://aclanthology.org/N19-1242.

[148] A. Sarkar, S. Reddy, and R. S. Iyengar. Zero-shot multilingual sentiment analysis using hierarchical attentive network and bert. In *Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval*, NLPIR 2019, pages 49–56, Tokushima, Japan. Association for Computing Machinery, 2019. ISBN: 9781450362795. DOI:

10.1145/3342827.3342850. URL: https://doi.org/10.1145/3342827.3342850.

[149] M. Birjali, M. Kasri, and A. Beni-Hssane. A comprehensive survey on sentiment analysis: approaches, challenges and trends. *Knowledge-Based Systems*, 226:107134, 2021. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2021.107134. URL: https://www.sciencedirect.com/science/article/pii/S095070512100397X.

[150] M. S. Rasooli, N. Farra, A. Radeva, T. Yu, and K. McKeown. Cross-lingual sentiment transfer with limited resources. *Machine Translation*, 32(1):143–165, 2018.

[151] A. Pelicon, M. Pranjić, D. Miljković, B. Škrlj, and S. Pollak. Zero-shot learning for cross-lingual news sentiment classification. *Applied Sciences*, 10(17), 2020. ISSN: 2076-3417. DOI: 10.3390/app10175993. URL: https://www.mdpi.com/2076-3417/10/17/5993.

[152] A. Kumar and V. H. C. Albuquerque. Sentiment analysis using xlm-r transformer and zero-shot transfer learning on resource-poor indian language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 20(5), June 2021. ISSN: 2375-4699. DOI: 10.1145/3461764. URL: https://doi.org/10.1145/3461764.

[153] P. Keung, Y. Lu, G. Szarvas, and N. A. Smith. The multilingual amazon reviews corpus. *arXiv preprint arXiv:2010.02573*, 2020.

[154] J. Kocoń, P. Miłkowski, and M. Zaśko-Zielińska. Multi-level sentiment analysis of PolEmo 2.0: extended corpus of multi-domain consumer reviews. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 980–991, Hong Kong, China. Association for Computational Linguistics, Nov. 2019. DOI: 10.18653/v1/K19-1092. URL: https://aclanthology.org/K19-1092.

[155] S. Smetanin and M. Komarov. Sentiment analysis of product reviews in russian using convolutional neural networks. In *2019 IEEE 21st Conference on Business Informatics (CBI)*, volume 01, pages 482–486, July 2019. DOI: 10.1109/CBI.2019.00062.

[156] V. Yadav and S. Bethard. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*, 2019.

[157] J. Li, A. Sun, J. Han, and C. Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2022. DOI: 10.1109/TKDE.2020.2981314.

[158] S. Ali, K. Masood, A. Riaz, and A. Saud. Named entity recognition using deep learning: a review. In *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*, pages 1–7. IEEE, 2022.

[159] A. Fritzler, V. Logacheva, and M. Kretov. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, pages 993–1000, Limassol, Cyprus. Association for Computing Machinery, 2019. ISBN: 9781450359337. DOI: 10.1145/3297280.3297378. URL: https://doi.org/10.1145/3297280.3297378.

[160] T. Moon, P. Awasthy, J. Ni, and R. Florian. Towards lingua franca named entity recognition with bert, 2019. DOI: 10.48550/ARXIV.1912.01389. URL: https://arxiv.org/abs/1912.01389.

[161] R. Hvingelby, A. B. Pauli, M. Barrett, C. Rosted, L. M. Lidegaard, and A. Søgaard. DaNE: a named entity resource for Danish. English. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4597–4604, Marseille, France. European Language Resources Association, May 2020. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.565.

[162] A. Jain, B. Paranjape, and Z. C. Lipton. Entity projection via machine translation for cross-lingual NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1083–1092, Hong Kong, China. Association for Computational Linguistics, Nov. 2019. DOI: 10.18653/v1/D19-1100. URL: https://aclanthology.org/D19-1100.

[163] B. Li, Y. He, and W. Xu. Cross-lingual named entity recognition using parallel corpus: a new approach using xlm-roberta alignment. *arXiv preprint arXiv:2101.11112*, 2021.

[164] S. Weber and M. Steedman. Zero-shot cross-lingual transfer is a hard baseline to beat in German fine-grained entity typing. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 42–48, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics, Nov. 2021. DOI: 10.18653/v1/2021.insights-1.7. URL: https://aclanthology.org/2021.insights-1.7.

[165] X. Pan, B. Zhang, J. May, J. Nothman, K. Knight, and H. Ji. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for

Computational Linguistics, July 2017. DOI: 10.18653/v1/P17-1178. URL: https://www.aclweb.org/anthology/P17-1178.

[166] A. Rahimi, Y. Li, and T. Cohn. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics, July 2019. URL: https://www.aclweb.org/anthology/P19-1015.

[167] M. Xiao and Y. Guo. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 119–129, Ann Arbor, Michigan. Association for Computational Linguistics, June 2014. DOI: 10.3115/v1/W14-1613. URL: https://aclanthology.org/W14-1613.

[168] J. Tiedemann. Cross-lingual dependency parsing with universal dependencies and predicted pos labels. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 340–349, 2015.

[169] J. Guo, W. Che, D. Yarowsky, H. Wang, and T. Liu. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1234–1244, Beijing, China. Association for Computational Linguistics, July 2015. DOI: 10.3115/v1/P15-1119. URL: https://aclanthology.org/P15-1119.

[170] O. Lacroix, L. Aufrant, G. Wisniewski, and F. Yvon. Frustratingly easy cross-lingual transfer for transition-based dependency parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1058–1063, 2016.

[171] M. Bansal. Dependency link embeddings: continuous representations of syntactic substructures. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 102–108, Denver, Colorado. Association for Computational Linguistics, June 2015. DOI: 10.3115/v1/W15-1514. URL: https://www.aclweb.org/anthology/W15-1514.

[172] D. Kondratyuk and M. Straka. 75 languages, 1 model: parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics, Nov. 2019. DOI: 10.18653/v1/D19-1279. URL: https://aclanthology.org/D19-1279.

[173] M. Ulčar and M. Robnik-Šikonja. Finest bert and crosloengual bert. In *International Conference on Text, Speech, and Dialogue*, pages 104–111. Springer, 2020.

[174] J. Nivre, M.-C. de Marneffe, F. Ginter, J. Hajič, C. D. Manning, S. Pyysalo, S. Schuster, F. Tyers, and D. Zeman. Universal Dependencies v2: an evergrowing multilingual treebank collection. English. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association, May 2020. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.497.

[175] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush. Transformers: state-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics, Oct. 2020. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6.

[176] L. Kellner. *Historical outlines of English syntax*. Macmillan, 1892.

[177] C. Dalton-Puffer. *The French influence on Middle English morphology: A corpus-based study on derivation*, volume 20. Walter de Gruyter, 2011.

[178] P. Durkin. *Borrowed words: A history of loanwords in English*. Oxford University Press, 2014.

[179] J. Eronen, M. Ptaszynski, F. Masui, A. Smywiński-Pohl, G. Leliwa, and M. Wroczynski. Improving classifier training efficiency for automatic cyberbullying detection with feature density. *Information Processing & Management*, 58(5):102616, Sept. 2021. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2021.102616. URL: https://www.sciencedirect.com/science/article/pii/S0306457321001126.

[180] G. Jawahar, B. Sagot, and D. Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics, July 2019. DOI: 10.18653/v1/P19-1356. URL: https://aclanthology.org/P19-1356.

Research Achievements

First Author Publications

1. J. Eronen, M. Ptaszynski, F. Masui, M. Arata, G. Leliwa, and M. Wroczynski. Transfer language selection for zero-shot cross-lingual abusive language detection. *Information Processing and Management*, 59(4):102981, 2022. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2022.102981. URL: https://www.sciencedirect.com/science/article/pii/S0306457322000978.

2. J. ERONEN, M. PTASZYNSKI, and F. MASUI. Comparing performance of different linguistically-backed word embeddings for cyberbullying detection. In *Proceedings of the 2021 International Workshop on Modern Science and Technology*, volume 2021, pages 169–173. The International Center of National University Corporation Kitami Institute of Technology, Oct. 2021.

3. J. Eronen, M. Ptaszynski, F. Masui, G. Leliwa, M. Wroczynski, M. Piech, and A. Smywinski-Pohl. Exploring the potential of feature density in estimating machine learning classifier performance with application to cyberbullying detection. In *The 7th Workshop on Linguistic and Cognitive Approaches to Dialog Agents (LaCATODA 2021) collocated with IJCAI 2021, CEUR Workshop Proceedings 2935*, pages 5–14, Aug. 2021. DOI: 10.48550/ARXIV.2206.01889. URL: http://ceur-ws.org/Vol-2935/paper1.pdf.

4. J. Eronen, M. Ptaszynski, F. Masui, G. Leliwa, M. Wroczynski, M. Piech, and A. Smywinski-Pohl. Initial study into application of feature density and linguistically-backed embedding to improve machine learning-based cyberbullying detection. In *The 6st Workshop on Linguistic and Cognitive Approaches to Dialog Agents (LaCATODA 2020) collocated with IJCAI 2020, Online*, Jan. 2021. DOI: 10.48550/ARXIV.2206.01889. URL: https://arxiv.org/abs/2206.01889.

5. J. Eronen, M. Ptaszynski, F. Masui, A. Smywiński-Pohl, G. Leliwa, and M. Wroczynski. Improving classifier training efficiency for automatic cyberbullying detection with feature density. *Information Processing & Management*, 58(5):102616, Sept. 2021. ISSN: 0306-4573. DOI: https://doi.org/10.1016/

j.ipm.2021.102616. URL: https://www.sciencedirect.com/science/article/pii/S0306457321001126.

Co-authored Publications

1. M. Ptaszynski, M. Zasko-Zielinska, M. Marcinczuk, G. Leliwa, M. Fortuna, K. Soliwoda, I. Dziublewska, O. Hubert, P. Skrzek, J. Piesiewicz, P. Karbowska, M. Dowgiallo, J. Eronen, P. Tempska, M. Brochocki, M. Godny, and M. Wroczynski. Looking for razors and needles in a haystack: multifaceted analysis of suicidal declarations on social media—a pragmalinguistic approach. *International Journal of Environmental Research and Public Health*, 18(22):11759, Nov. 2021. ISSN: 1660-4601. DOI: 10.3390/ijerph182211759. URL: http://dx.doi.org/10.3390/ijerph182211759.
2. M. Ptaszynski, J. K. K. Eronen, and F. Masui. Learning deep on cyberbullying is always better than brute force. In *LaCATODA 2017 CEUR Workshop Proceedings*, pages 3–10, 2017. URL: http://ceur-ws.org/Vol-1926/paper1.pdf.