Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Rok Marinšek

Uporaba medjezičnih vektorskih vložitev za odkrivanje sovražnega govora v komentarjih

magistrsko delo

magistrski program druge stopnje
računalništvo in informatika

Mentor: prof. dr. Marko Robnik-Šikonja
Somentor: Prof. Dr. Alexander M. Fraser

Ljubljana, 2019

University of Ljubljana
Faculty of Computer and Information Science

Rok Marinšek

Cross-lingual embeddings for hate speech detection in comments

master's thesis

the 2nd cycle master's study programme
computer and information science

Supervisor: prof. dr. Marko Robnik-Šikonja
Co-supervisor: Prof. Dr. Alexander M. Fraser

Ljubljana, 2019

Acknowledgments

I would like to express my extreme gratitude to my supervisors, prof. dr. Marko Robnik-Šikonja, who has been extremely responsive when I had questions, steering me in the right direction, and giving suggestions throughout the writing of the thesis, and Prof. Dr. Alexander Fraser, for his guidance and suggestions during my time at Ludwig-Maximilians-Universität München. I would also like to thank my family and friends, who have supported me during my studies and time spent abroad.

It would not have been possible without them. Thank you!

Rok Marinšek, 2019

Contents

Povzetek

Naslov: Uporaba medjezičnih vektorskih vložitev za odkrivanje sovražnega govora v komentarjih

V zadnjih letih se je z eksplozijo vsebin na družbenih medijih povišala količina sovražnega govora. Zaradi velike količine podatkov je ročno moderiranje sovražnih vsebin nemogoče. Trenutno za avtomatsko odkrivanje sovražnega govora najpogosteje uporabljamo nevronske mreže. Za njihovo učenje je potrebno veliko označenih primerov, ki so večinoma na voljo le za večje jezike, npr. za angleščino. Označenih podatkov za manjše jezike je načeloma malo. Vseeno bi želeli tudi v teh jezikih zaznavati sovražni govor. V tem delu s pomočjo medjezikovnih vložitev razvijemo metodo, ki ob prenosu dosega sprejemljive rezultate za ciljni jezik. Komentarji s sovražnim govorom so v angleščini, nemščini in hrvaščini. Uporabimo fastText vložitve, jih poravnamo z metodo RCSLS, in dosežemo sprejemljive rezultate za dva od šestih jezikovnih parov. Z modelom BERT izboljšamo to metodo in dosežemo sprejemljive rezultate za tri od šestih jezikovnih parov.

Ključne besede

vektorska vložitev, medjezikovna vložitev, globoko učenje, odkrivanje sovražnega govora, obdelava naravnega jezika, metoda RCSLS, jezikovni model BERT

Abstract

Title: Cross-lingual embeddings for hate speech detection in comments

With the recent explosion of social media content, the amount of online hate speech is increasing, making it impossible to filter it manually. For automatic hate speech detection, a lot of annotated data is needed, which is mostly available for high-resource languages. In spite of data scarcity in low-resource languages, we want to detect hate speech in those languages. We use cross-lingual embeddings to achieve an acceptable performance in hate speech detection in a target language, using data from another language. We use hate speech comments from English, German, and Croatian. We use fastText word embeddings, align them with the RCSLS method, and achieve reasonable performance in 2 out of 6 language pairs. With Multilingual BERT, we improve upon this method, and achieve acceptable performance in 3 out of 6 language pairs.

Keywords

word embedding, cross-lingual embedding, deep learning, hate speech detection, natural language processing, RCSLS method, BERT language model

Razširjeni povzetek

V zadnjih letih se je z eksplozijo vsebin na družbenih medijih povišala količina sovražnega govora. Zaradi velike količine podatkov je ročno moderiranje sovražnih vsebin nemogoče. Trenutno za avtomatsko odkrivanje sovražnega govora najpogosteje uporabljamo nevronske mreže. Za njihovo učenje je potrebno veliko označenih primerov, ki so večinoma na voljo le za večje jezike, npr. za angleščino. Označenih podatkov za manjše jezike je načeloma malo. Vseeno bi želeli tudi v teh jezikih zaznavati sovražni govor.

V tem delu s pomočjo medjezikovnih vložitev razvijemo metodo, ki ob prenosu dosega sprejemljive rezultate za ciljni jezik.

I Kratek pregled sorodnih del

Na področju odkrivanja sovražnega govora večinoma poskušamo razločiti sovražni in nesovražni govor [33]. Pogosto za odkrivanje sovražnega govora uporabljamo SVM in nevronske mreže. Med nevronskimi mrežami so konvolucijske mreže in rekurentne nevronske mreže najbolj priljubljene [4, 14, 27, 31, 33].

Da bi nevronske mreže lahko učili s tekstovnimi podatki, moramo besedila pretvoriti v številske vrednosti, za kar uporabljamo vektorske vložitve. Pomembna značilnost vektorskih vložitev je, da so besede, ki so si pomensko blizu, blizu tudi v vektorskem prostoru [21].

Dve znani vložitvi sta word2vec [21] in GloVe [23]. FastText [7] je izboljšana inačica metode word2vec [21], vendar ne upošteva konteksta, torej

homografi zasedajo v vektorskem prostoru isto točko. Metoda ELMo [24] upošteva kontekst tako, da je vsaka beseda predstavljena kot funkcija celotnega stavka.

Isti pomeni v različnih jezikih zavzemajo različna mesta v vektorskih prostorih. Če bi želeli podatke iz različnih jezikov uporabiti za skupno klasifikacijo, bi morali biti v skupnem prostoru. V ta namen so bile razvite medjezikovne vložitve.

Conneau et al. [10] so pokazali, da lahko sestavimo dvojezični slovar brez uporabe paralelnih korpusov. Razvili so metriko CSLS (angl. cross-domain similarity local scaling), ki poskuša najti poravnavo dveh jezikov tako, da je najbližji sosed besede v izvornem jeziku najbližji sosed besede tudi v ciljnem jeziku. Joulin et al. [19] izboljšajo metriko CSLS z metriko RCSLS (angl. relaxed CSLS). Pokazali so, da je, če sprostimo omejitev ortogonalnosti pri učenju preslikave, metrika RCSLS konveksna in jo lahko minimiziramo. S to kriterijsko funkcijo so, predvsem za jezike, ki si niso blizu (npr. angleščina-kitajščina), izboljšali rezultate metode CSLS.

II Predlagana metoda

Glavna ideja naše metode je, da model naučimo na večji podatkovni množici v izvornem jeziku, ki je poravnan s ciljnim jezikom, model potencialno doučimo na manjši množici v ciljnem jeziku in klasificiramo sovražni govor v ciljnem jeziku.

Za poravnavo smo uporabili metodo RCSLS in učne slovarje s 5.000 besednimi pari. Opazimo, da sta para angleščina-nemščina in nemščina-angleščina dobro poravnana. Jezikovni pari s hrvaščino so mnogo slabše poravnani.

Podatkovne množice v izvornem jeziku so večje (največ 10.000 primerov) in vsebujejo približno 10 % primerov sovražnega govora. Za ciljni jezik uporabimo tri podatkovne množice: učno, validacijsko in testno. Te množice so manjše (do 2.000 primerov) in vsebujejo približno 50 % primerov sovražnega

govora. Pri evalvaciji metode ostaneta validacijska in testna množica konstantni, spreminjamo le učno množico.

Za klasifikacijo smo razvili dve nevronski mreži; CNN in BiLSTM-CNN. Struktura konvolucijske mreže (CNN) temelji na mreži, ki jo je razvil Kim [20]. S to mrežo želimo prepoznati vzorce, ne glede na to, kje se pojavijo. Z mrežo BiLSTM-CNN želimo s slojem BiLSTM zajeti relacije med besedami, in s konvolucijskim slojem zajeti lokalne značilke.

Za testirane modele poročamo njihovo točnost, priklic in mero F1. Da bi razumeli vpliv primerov v izvornem jeziku na klasifikacijsko točnost v ciljnem jeziku, izračunamo tudi korelacijo med številom primerov v ciljnem/izvornem jeziku in metrikami. S podobnim namenom dodajamo izvornim podatkom podatke v ciljnem jeziku in opazujemo dodano vrednost izvornih podatkov.

III Eksperimentalna evalvacija

Rezultate lahko glede na kakovost preslikav razdelimo v dve skupini. V prvi skupini sta jezikovna para angleščina-nemščina in nemščina-angleščina, ki sta dobro poravnana in dosegata dobre rezultate. V drugi skupini so jezikovni pari s hrvaščino, ki so slabo poravnani in dosegajo slabše rezultate.

Jezikovni par angleščina-nemščina dosega najboljše rezultate. Opazimo, da imajo primeri v izvornem jeziku v vsaki testirani kombinaciji dodano vrednost. Predvsem pri učenju brez primerov v ciljnem jeziku dosegamo sprejemljive rezultate. V korelacijski tabeli (tabela 6.2) opazimo, da imajo izvorni primeri primerljiv vpliv na klasifikacijsko točnost kot primeri v ciljnem jeziku. Ugotovimo, da imajo izvorni primeri pri dobrih poravnavah primerljiv vpliv na klasifikacijsko točnost kot ciljni primeri.

V skupini s slabšimi poravnavami sta dva zanimivejša jezikovna para. Jezikovni par hrvaščina-nemščina je najboljši primer slabe poravnave, par hrvaščina-angleščina pa je najslabši primer slabe poravnave. Izvorni primeri pri paru hrvaščina-nemščina so pozitivno korelirani z mero F1 (+0,11), pri paru hrvaščina-angleščina pa negativno korelirani (−0,11). Ugotovimo, da

Tabela 1: Primerjava rezultatov metode RCSLS in modela BERT. Učenje samo s primeri v ciljnem jeziku (tgt) in samo s primeri v izvornem jeziku (src). Poročamo priklic (r), točnost (p), mero F1 (f) za najboljši rezultat (obeh modelov). Krepko besedilo označuje najboljši rezultat stolpca.

	en-de			en-hr			de-en			de-hr			hr-en			hr-de		
	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f
BERT tgt	0,72	**0,71**	**0,72**	**0,74**	**0,74**	**0,74**	**0,88**	0,72	0,79	0,71	**0,75**	**0,73**	0,85	0,73	**0,78**	0,73	0,69	0,71
RCSLS tgt	**0,75**	0,68	0,70	0,65	0,72	0,70	0,75	**0,84**	**0,80**	**0,72**	0,70	0,71	**0,90**	0,73	**0,78**	**0,79**	**0,73**	**0,76**
BERT src	0,65	0,64	0,64	0,49	0,58	0,53	0,63	0,67	0,65	0,67	0,61	0,64	0,77	0,64	0,70	0,71	0,53	0,61
RCSLS src	0,45	0,70	0,62	0,24	0,64	0,51	0,63	0,71	0,69	0,19	0,59	0,47	0,11	**0,84**	0,44	0,03	0,44	0,37

imajo izvorni primeri v primeru slabih poravnav zanemarljiv oziroma negativen vpliv na klasifikacijsko točnost v ciljnem jeziku.

Zaradi slabših rezultatov parov s hrvaščino smo se odločili testirati še model BERT. Ugotoviti želimo, če je za slabe rezultate odgovorna slaba poravnava. Modelu BERT doučimo vse skrite sloje s podatki v angleščini, nemščini in hrvaščini. Na njegovem izhodu uporabimo enaka klasifikatorja kot pri metodi RCSLS. V tabeli 1 primerjamo rezultate metode RCSLS in modela BERT. Opazimo, da BERT večinoma izboljša rezultate metode RCSLS in doseže sprejemljive rezultate pri treh od šestih jezikovnih parov, če učimo samo s primeri v izvornem jeziku. Izrazite so izboljšave pri parih nemščina-hrvaščina, hrvaščina-angleščina in hrvaščina-nemščina. Zaradi tega sklepamo, da so bile slabe poravnave vzrok za slabe rezultate pri parih s hrvaščino.

IV Sklep

Naša prvotna hipoteza je bila, da bomo z medjezikovnimi vložitvami prenesli nekaj znanja iz izvornega v ciljni jezik. Pokazali smo, da lahko samo s podatki v izvornem jeziku dosežemo sprejemljive rezultate pri prepoznavi sovražnega govora v ciljnem jeziku, vendar je za to potreben dobro poravnan skupen prostor. Z modelom BERT smo izboljšali rezultate metode RCSLS; predvsem pri učenju samo s primeri v izvornem jeziku so izboljšave izrazite. Ugotovimo, da naj bo to prva izbira, ko želimo jezikovne prenose znanja.

Omejitev našega pristopa je, da naši modeli uporabljajo privzete parametre in ne upoštevajo izvenslovarskih besed. Sklepamo, da bi z optimizacijo parametrov in upoštevanjem izvenslovarskih besed dosegli boljše rezultate.

Čeprav smo se pri izbiri podatkovnih množic sovražnega govora trudili najti tematsko čim bolj podobne, so naše učne množice vseeno tematsko različne. Angleška (forum Stormfront) in nemška (npr. Facebook skupina Pegida) sta specifični, medtem ko je hrvaška množica splošna. Sklepamo, da tematska različnost dodatno slabša rezultate.

Chapter 1

Introduction

With the explosion of social media content in recent years, the amount of online hate speech is increasing. Since there is no consensus concerning the definition of hate speech, it is difficult to identify. Legal and academic literature generally define it as speech that expresses hatred against a person or group of people because of a characteristic they share, or a group to which they belong [26].

Companies like Twitter, Facebook and Youtube have been investing millions of euros every year into moderating hate speech on their platforms, but are still criticised for not doing enough. Currently, most of the efforts to identify and delete offensive posts are done manually. As such, the process is extremely labour intensive, time consuming, and not sustainable or scalable. Given the massive amount of data, it has become impossible to manually process and detect potential hate speech. Thus, the need for automated hate speech detection arose [33].

Currently, most methods for hate speech detection use supervised learning. Since hate speech lacks unique, discriminative features, best results are achieved using deep learning [27, 31, 33]. For that approach to work best, a lot of training data is needed, which may be costly. Construction of large datasets in low-resource languages poses a further challenge. We would like to achieve good classification results exploiting existing resources in other

languages.

In recent years, cross-lingual word embeddings have enabled us to reason about word meaning in multilingual contexts and are a key facilitator of cross-lingual transfer when developing natural language processing (NLP) models for low-resource languages [25]. In this thesis, we propose an approach using cross-lingual embeddings for solving the issue of low-resource languages in the context of hate speech detection. We show that using cross-lingual embeddings and data from other languages can compensate for the lack of data in the target language.

This thesis is organised as follows: In Chapter 2, we give an overview on related work done in the field of hate speech detection. In Chapter 3, we take a closer look at the used datasets. We have chosen datasets in English, German, and Croatian. In Chapter 4, we review the ideas behind word and cross-lingual embeddings. In Chapter 5, we present our approach, our implementation and models used for classification. In Chapter 6, we present and comment on the results achieved for the six language combinations tested. We conclude with Chapter 7 where we summarise our work and present implications. We analyse shortcomings of our approach and discuss possible improvements.

Chapter 2

Related work on hate speech detection

In the UK, there has been a significant increase in hate speech towards migrant and Muslim communities following events like leaving the EU or the Manchester and London attacks. In the EU, surveys and reports focusing on young people show an increase of hate speech and related crimes based on religion, sexual orientation, or gender. Statistics also show a similar trend in the US since the Trump election. A range of international initiatives have been launched towards the qualification of the problems and the development of counter measures [33].

In this section, we present work done on automated hate speech detection. In recent years, extensive research has been conducted to develop automated methods for hate speech detection on social media. These typically employ semantic content analysis techniques based on natural language processing (NLP) and machine learning methods. Usually they try to distinguish hate from non-hate content. Although this usually shows good results, evaluations are often biased towards detecting non-hate, as opposed to detecting hateful content. In some domains, detecting hate speech achieves between 15 and 60 percentage points lower F1 scores, as opposed to detecting non-hate speech. This suggests that detecting hate speech is a much harder task [33].

Some of the techniques which often achieve good results are support-vector machines (SVM) and deep learning [4, 14, 27, 31]. SVM tries to find a line or hyperplane which separates the data into classes. It relies on manually designing and encoding features of data samples into feature vectors, which are then used by the classifier. Deep learning methods learn abstract feature representations of features from input data through multiple stacked layers, which means that the input features may not be used for classification. The two most popular neural network architectures are convolutional neural network (CNN) and recurrent neural network (RNN). In the context of hate speech, CNNs extract word or character combinations, whereas RNNs learn word or character dependencies [33]. Zhang and Luo [33] have outperformed other state of the art models using a skipped CNN.

Most research is focused on hate speech in English. Some examples of English datasets are presented in Davidson et al. [11] and de Gibert et al. [12]. Examples of other languages where research is done include German [8] and Italian [30]. We have found no examples of research for low-resource languages.

Davidson et al. [11] note that one of the problems in detecting hate speech is distinguishing it from offensive speech. Building on that, Waseem [32] shows the importance of labelling the datasets correctly. It is important to note that amateur annotators are more likely to label items as hate speech and systems trained on expert annotations outperfrom systems trained on amateur annotations.

Zhang and Luo [33] show that hateful content is usually found in the long tail of a dataset (most hate speech samples occur far away from the central part of the distribution) due to their lack of unique, discriminative features. Due to that, the practice of 'micro-averaging' over both hate and non-hate classes can be questionable, since it can be significantly biased towards the dominant non-hate class and overshadow the method's ability to detect hateful content.

Chapter 3

Datasets

In this thesis, we have chosen datasets from three languages: English, German, and Croatian. Even though English and German are representatives of high-resource languages, Croatian is the language where the available dataset is magnitudes larger. In the context of hate speech, even high-resource languages do not seem to have many publicly available datasets, as the largest have at most 15,000 labelled samples. Datasets from different hate domains introduce further uncertainty.

3.1 English dataset

For the English dataset, created by de Gibert et al. [12], the data was scraped from Stormfront, the largest online community of white nationalists and known as the first hate website. The extracted content was published between 2002 and 2017 and presents the first dataset of textual hate speech annotated at sentence-level. Sentence-level annotation allows to work with the minimum unit containing hate speech and reduce noise introduced by other sentences that are clean. They also include data about how much context the annotator needed to classify each sentence. Annotators discussed guidelines to ensure everyone had the same understanding of hate speech.

As seen in Table 3.1, a total of 10,703 sentences have been extracted

Table 3.1: The English dataset distribution of hate speech samples [12].

	#samples	%
noHate	9,507	88.83
hate	1,196	11.17
total	10,703	100.00

Table 3.2: Two samples from the English dataset [12].

comment	label
That's all I needed to hear .	0
He is a pathetic little chimp .	1

from Stormfront and classified as hate speech or not. 1,196 samples have been labelled as hate speech and 9,507 as non-hate speech.

In Table 3.2, two samples from the dataset are shown. All samples are labelled with either 0 or 1, 0 denoting a neutral comment and 1 a hateful comment. In some samples, added context was needed to label the comments. This hints that those samples marked as hateful are noisier than usual.

The authors of the dataset tested three models, SVM, CNN, and long short-term memory (LSTM), all with no special optimisation. They tested the models on a balanced subset of 2,000 labelled sentences. From this subset, 80% are used for training and the remaining 20% for testing. The model structures are constructed in the following way: SVM uses bag-of-words vectors, CNN is a simplified version of Kim's model [20] using a single input channel of randomly initialised word embeddings, and the LSTM with a layer size of 128 over word embeddings of size 300. As seen in Table 3.3, LSTM performed best, but SVM using bag-of-words vectors was close to its performance. The CNN model performed worse than both. Instead of choosing the recall and precision metric for a single class, the authors decided to

Table 3.3: The English dataset model performance [12].

model	ACC_{HATE}	ACC_{NOHATE}	ACC_{ALL}
SVM	0.69	0.73	0.71
CNN	0.54	0.79	0.66
LSTM	0.71	0.75	0.73

report sensitivity and specificity to highlight the performance for both classes individually [12].

3.2 German dataset

The German dataset by Bretschneider and Peters [8] was constructed by accessing publicly available Facebook pages. Three pages were chosen. "Pegida" (dataset 1) and "Ich bin ein Patriot, aber kein Nazi" (dataset 2) are known for their critical view of foreigners and refugees. "Kriminelle Ausländer raus" (dataset 3) is known for xenophobe and racist comments. Two human experts annotated the datasets marking offensive statements, their severity and intended target. Each offending passage is marked and assessed with a severity value. Statements perceived by the experts as slightly offensive to offensive are denoted with a severity value of 1, and explicit to substantial offensive statements with a value of 2.

Since we want to predict only two classes (neutral or hate), we have made a new dataset where a sample with severity at least 1 was given the label 1. In Table 3.5, two samples from the German dataset are shown. All samples are labelled either 0 or 1, with 0 denoting a neutral comment and 1 a hate comment.

Bretschneider and Peters [8] tested two models on the dataset. Their pattern-based approach uses an architecture which detects references and recognises hate speech patterns. As can be seen in Figure 3.1, their pattern-

Table 3.4: The German dataset summary [8]. Cohen's kappa measures inter-rater agreement. As it takes agreement by chance into account, it is a more robust agreement measure than percent agreement calculation.

Dataset	1	2	3	total
#comments	2.649	2.641	546	5.836 (100.00%)
#cases (severity=1)	99	112	50	261 (4.47%)
#cases (severity=2)	137	112	130	379 (6.49%)
Cohen's kappa	0.78	0.68	0.73	0.73

Table 3.5: Two samples from the German dataset [8].

comment	label
Bla bla bla bla	0
Ausländer sind alle Dreck	1

Figure 3.1: The German dataset model performance [8]. The reported evaluation scores are precision (p), recall (r) and F1 score (f1).

	Dataset 1			Dataset 2		
	p	r	f1	p	r	f1
Baseline	53.57	76.27	62.94	50.65	71.43	59.27
Pattern-based	75.26	61.86	67.91	73.89	53.46	62.03

based approach performed better than the tested baseline model using a bag-of-words approach [8]. To assess performance of the binary classification they report precision (p), recall (r) and the F1 score (f1).

3.3 Croatian dataset

The Croatian dataset is provided for research purposes within the EMBEDDIA project [2]. There are two files of user comments extracted, one from the 24sata.hr news portal and the other from vecernji.hr. Both datasets have 11 columns. The 24sata dataset has 21,548,192 rows and Vecernji List dataset has 9,646,634 rows, where each row represents one user comment.

Before we could use the data, we had to do some preprocessing. One of the columns is "infringed_on_rule", which has data about the infringement type of the comment (if there has been an infringement). The relevant infringements for us are "Rule ID 3" in 24sata dataset, which is given when the comment contains "Verbal abuse, derogation and verbal attack based on national, racial, sexual or religious affiliation, hate speech and incitement". For the Vecernji List dataset "Rule ID 1" is relevant, which marks a comment that contains "Hate speech on a national, religious, sexual or any other basis". We have taken those samples and marked them as hate speech (label 1). Comments that do not infringe on any rules, we consider neutral comments and mark them with 0. We disregarded all other comments that do not fall in those two categories e.g. political trolling, verbal abuses of moderators etc. We extracted 84,509 hate speech comments from the Vecernji List

Table 3.6: Two samples from the Croatian dataset.

comment	label
tko to gleda	0
hahahaha!	1

dataset and 19,304 from the 24sata dataset. Given that the 24sata dataset is about twice the size of Vecernji List and contains only 19% of the hate speech comments, it is fair to assume that the annotators had different standards for annotating the comments. Two examples of the extracted comments can be seen in Table 3.6. From the positive example, we can see that the labelling is context dependent since the comment itself cannot be considered hate speech. Upon further inspection, it was deemed hateful because it was a response to a hateful comment and seen as a support of hate speech. we noticed multiple samples like that, and assume that such comments will be difficult to recognise as hate speech. By removing comments marked as replies, we could reduce this issue.

We created two datasets (see Table 3.7) with the data subset (neutral and hateful comments) to use for model training. The "hr_full" dataset contains all samples combined from the two before-mentioned datasets (24sata and Vecernji list). To ensure fixed samples during evaluation, we made train, validation and test splits of "hr_source_12k" and "hr_bal_3k". The "hr_source_12k" dataset contains 12,000 samples and hate speech ratio of 0.11. This is similar in terms of size and hate speech ratio to the datasets in English and German. It is used as training set for Croatian as the source language. A balanced dataset ("hr_bal_3k"), which has a hate speech ratio of about 50%, is used as a training set for Croatian as the target language.

Table 3.7: Croatian datasets summary.

Dataset	#non-hate	#hate	hate ratio	total
hr_full	30,293,479	103,813	0.003	30,397,292
24sata_full	21,094,135	19,304	0.001	21,113,439
vecernji_full	9,199,344	84,509	0.01	9,283,853
hr_source_12k	10,680	1,320	0.11	12,000
hr_train_12k	6,826	857	0.11	7,683
hr_val_12k	1,710	218	0.11	1,928
hr_test_12k	2,114	245	0.11	2,389
hr_bal_3k	1,500	1,500	0.50	3,000
hr_bal_train	960	971	0.50	1,931
hr_bal_val	227	221	0.49	448
hr_bal_test	313	308	0.50	621

Chapter 4

Word embeddings

An important part of this thesis is how we represent words as numeric values. Word embeddings give us a vector representation of a word in an embedded space. It has been shown that words similar in meaning are close to each other in an embedded vector space. This is the main idea behind our thesis. We assume that hate speech will be represented similarly in an embedded space.

Different languages have semantics presented in different areas of the vector space. However, to use them for classification, they should be sharing the same space. To achieve a shared space, cross-lingual embeddings have been developed. Cross-lingual embeddings enable better insight into word meanings in multilingual contexts. In Figure 4.1, we see an example of aligning two vector spaces. Our assumption is that we can use this characteristic of cross-lingual embeddings to detect hate speech independent of language.

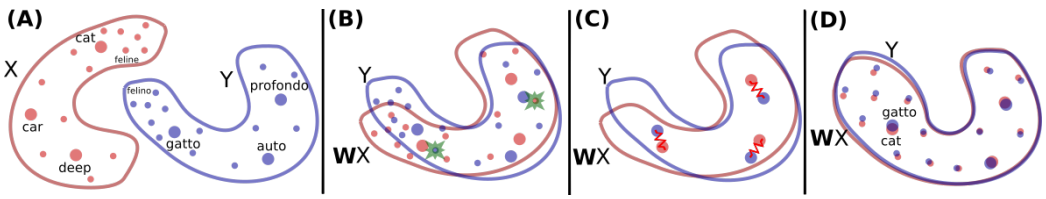

Figure 4.1: Cross-lingual alignment of a vector space [10].

4.1 Review of embeddings

Mikolov et al. [21] introduce a Skip-gram model for learning vector representations of words from unstructured data. Training Skip-gram model does not involve dense matrix multiplications. This makes training efficient. An interesting property of the Skip-gram model is that simple vector addition can often produce meaningful results. For example, vec("Germany") + vec("capital") is close to vec("Berlin") (see Figure 4.2). This suggests that a non-obvious degree of language understanding can be obtained by using basic mathematical operations on word vector representations.

Two well-known vector embeddings are word2vec [21] and GloVe [23]. FastText builds upon word2vec with subword information and outperforms baselines that do not take subword information into account [7]. In word embeddings, fastText offers a strong baseline, but can be improved upon with ELMo [24], where each word is represented as a function of the entire sentence. Similar to fastText, ELMo also takes advantage of subword information and thus can compute meaningful representations of out-of-vocabulary (OOV) words.

In cross-lingual contexts, we have three different types of alignments: word, sentence and document alignment. Most approaches use word-aligned data in the form of bilingual or cross-lingual dictionary with pairs of translations. Sentence alignment requires a parallel corpus that is commonly used for machine translation. Document alignment is the rarest of the three since it requires parallel document-aligned data in different languages that are translations of each other. One of the challenges that is relevant also to us is embeddings for specialised domains where parallel data is scarce. That makes robust cross-lingual word representations with as few parallel examples as possible an important research direction [25] .

Conneau et al. [10] have shown that we can build a bilingual dictionary between two languages without using any parallel corpora, by aligning monolingual word embedding spaces in an unsupervised way. Using adversarial training, the authors are able to find a linear mapping between a source and

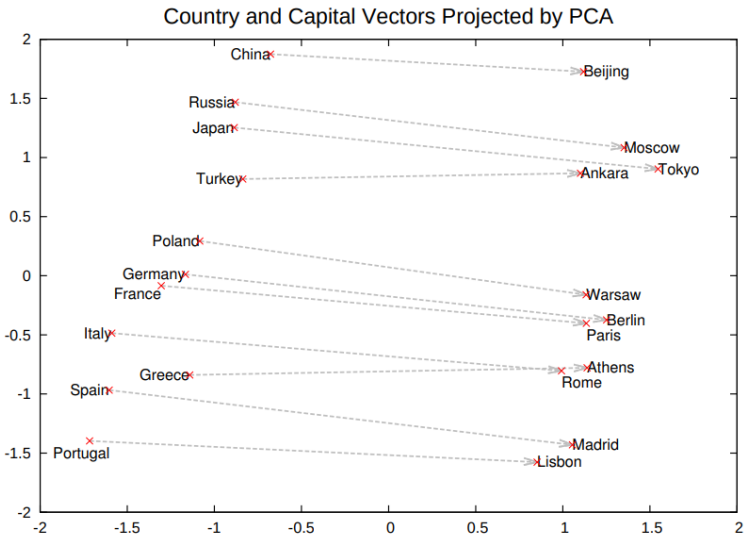

Figure 4.2: word2vec representations of countries and capitals in 2D space [21].

target space, which they use to produce a parallel dictionary. In some cases, their approach outperforms the quality of supervised approaches. They introduced a new comparison metric called cross-domain similarity local scaling (CSLS). It aims to produce reliable matching pairs between languages such that the nearest neighbour of a source word, in the target language, is more likely to have this particular word as a nearest neighbour. Glavas et al. [15] question the used metrics when measuring quality of cross-lingual embeddings, claiming CSLS is prone to overfitting to bilingual lexicon induction (BLI). Joulin et al. [19] improve upon MUSE [10] with the relaxed CSLS (RCSLS) method. They show that minimising a convex relaxation of the CSLS loss significantly improves the quality of bilingual word vector alignment.

Contextualised word embeddings can also be aligned. One of the problems of cross-lingual alignments is that each token pair is represented by many different vectors depending on its context. Even when supervision is available in the form of a dictionary, it is unclear how to utilise the multi-

ple contextualised embeddings so that they correspond to a word translation pair. One of the approaches that bypasses the problem and works well in low-resource environments is presented by Schuster et al. [28]. Instead of learning alignments in the original contextual space, the mapping process uses context-independent embedding anchors. Anchors are obtained by factorising the contextualised embeddings space into context-independent and context-dependent parts. This enables us to utilise a word-level dictionary as a source of supervision. Once the anchor-level alignment is learned, it can be applied to map the original spaces with contextualised embeddings.

4.2 Alignment with RCSLS method

Currently, most aligning methods solve a least-square regression problem to learn a rotation aligning a small bilingual lexicon, and use a retrieval criterion for inference. However, most of the models suffer from the "hubness problem": some word vectors tend to be nearest neighbours of a large number of other words. Conneau et al. [10] solved the problem by applying a corrective metric at inference time called CSLS. The CSLS criterion (Equation 4.1) is a similarity measure between the vectors x and y defined as:

$$CSLS(x, y) = -2\cos(x, y) + \frac{1}{k}\sum_{y' \in \mathcal{N}_Y(x)} \cos(x, y') + \frac{1}{k}\sum_{x' \in \mathcal{N}_X(y)} \cos(x', y) \quad (4.1)$$

where $\mathcal{N}_Y(x)$ is the set of k nearest neighbours of the point x in the set of target word vectors $Y = \{y_1, ..., y_N\}$, and cos is the cosine similarity. This is not totally satisfactory because the loss used in inference (CSLS to infer word correspondences) is not consistent with the one used in training (square loss to find a orthogonal mapping W, which suffers from the existence of "hubs").

Joulin et al. [19] propose an unified formulation that directly optimises a retrieval criterion in an end-to-end fashion. Their training objective is inspired by the CSLS retrieval criterion. Convex relaxations of the corresponding objective function are introduced, which are optimised with projected subgradient descent.

The optimisation problem with the relaxed CSLS (RCSLS) loss is written as:

$$\begin{aligned} RCSLS(x,y) = \min_{W \in \mathcal{O}_d} \frac{1}{n} \sum_{i=1}^{n} & -2x_i^T W^T y_i \\ & + \frac{1}{k} \sum_{y_j \in \mathcal{N}_Y(Wx_i)} x_i^T W^T y_j \\ & + \frac{1}{k} \sum_{Wx_j \in \mathcal{N}_X(y_i)} x_j^T W^T y_i \,, \end{aligned} \tag{4.2}$$

where the linear mapping W is constrained to belong to the set of orthogonal matrices $\mathcal{O}_d$. We also assume the word vectors are ℓ_2-normalised. Under these assumptions, $\cos(Wx_i, y_i) = x_i^T W^T y_i$ is true. Similarly, we have $\|y_j - Wx_i\|_2^2 = 2 - 2x_i^T W^T y_j$. This means that finding k nearest neighbours of Wx_i among the elements of Y is equivalent to finding k elements of Y which have the largest dot product with Wx_i. When relaxing the orthogonality constraint on W, we get a convex formulation of the loss, since the second and third term can be written as a function of W, maximum of linear function of W, which is convex. This means that we can minimise this objective function by using projected subgradient descent.

Using this objective function, the authors achieve a significant quality improvement of bilingual word vector alignment. Especially improvements in distant languages (like English-Chinese) have been significant [19]. This motivates our use of this method for alignments.

4.3 BERT language model

Currently, one of the state-of-the-art models for language representation is BERT [13]. It is based on transformer neural architecture [29], which have certain advantages over sequential models, e.g., LSTM. A transformer is an encoder-decoder architecture which uses attention mechanisms to feed the whole sequence to the decoder at once, instead of sequentially like LSTM. This allows for more effective modeling of long-term dependencies.

The BERT model is built with bidirectional transformers using encoders. It employs masked language modeling (MLM) as a training objective, which means that 15% of words are hidden and their position is used to infer them. The base BERT model has 12 hidden transformer layers trained on a large general corpus. Multilingual BERT is trained with corpora from 104 languages and enables cross-lingual knowledge transfer [13].

Because of the large model size, training BERT is computationally expensive. That is why we use pre-trained models. To use pre-trained BERT models for classification, we usually fine-tune all the hidden layers and add a softmax layer for classification. This methodology achieves state-of-the-art performance in many tasks [13].

Chapter 5

Cross-lingual embeddings for hate speech detection

In this chapter, we present our approach of using cross-lingual embeddings to detect hate speech. As previously mentioned, the idea is to use a bigger dataset in a source language and through cross-lingual embeddings predict samples in a smaller dataset in the target language. To see the added value of source samples, we add different amounts of target samples. Our approach consists of these main steps:

1. Align source and target language vector space (Section 5.1).
2. Prepare source and target data (Section 5.2).
3. Build and train neural net with source and target samples (Section 5.3).
4. Evaluate the models and report results (Section 5.4).

All code is written in Python. For alignment, we have used the code and vectors provided by Joulin et al. [19]. They offer fastText vectors as an effective word embedding method and a tool to transform a source and target language into an aligned common space. To do the data processing, we have used the libraries NumPy, Pandas [18], and NLTK [6]. To build, train and

evaluate the models, we have used Keras [9] and sklearn [22], for plotting we used matplotlib [17]. After classification, we evaluate and report our results. Where sensible, we compare them to the results of the original authors.

5.1 Aligning embeddings

To align our pretrained fastText embeddings for all language pairs, we have used the RCSLS method presented in Section 4.2. The training dictionary has 5,000 word pairs and the test dictionary has 1,500 word pairs. Since cross-lingual alignments contain a certain amount of error, we report the quality of the alignments in Table 5.1. As expected, English and German are better aligned than Croatian. This confirms that languages that are closer to each other have better alignments. Our alignments achieve lower scores than the same language combinations presented in Joulin et al. [19].

Since we did not find a Croatian-German and German-Croatian dictionary, we had to make them ourselves. We used English dictionaries which are available in both directions for both languages. For example, to get the German-Croatian dictionary, we merged the pairs from the German-English and English-Croatian dictionary, where the English entry was equal. The alignment metrics for the pairs German-Croatian and Croatian-German are poor, which could be due to the self-constructed dictionary. The entries chosen for the dictionary may not be suitable as anchors for the alignment. Another possibility is that the test dictionary is not well chosen.

5.2 Datasets

In this section, we describe the data we embedded using the aligned embeddings described in the previous section. For every language combination tested, we have used datasets described in Section 3. We did some preprocessing, we made the text lower case, removed unnecessary whitespaces, removed URLs and mentions (strings that begin with @). We tokenised the

Table 5.1: Alignment metrics for all language combinations. The metrics chosen are nearest neighbour (NN), CSLS, and coverage, which respresents the ratio of covered samples in the testing dictionary.

Combination	NN	CSLS	Coverage
en - de	0.71	0.74	1
en - hr	0.33	0.37	1
de - en	0.71	0.74	1
de - hr	0.12	0.17	1
hr - en	0.46	0.49	1
hr - de	0.28	0.35	1

text and embedded the tokens in the vector space.

From here on, we call the dataset in the source language the source dataset and the dataset in the target language the target dataset. Since the datasets are heavily imbalanced, we made smaller target datasets with ≈50% neutral and ≈50% hate samples. During preliminary testing, this approach showed better performance in detecting hate speech. This also has two side benefits, for one, we emphasise our models ability to detect hate speech, and since we make small datasets, we simulate a low-resource environment. Consequently, the source dataset is imbalanced and the target dataset is our balanced during training. An exception to this is the German dataset, where using a balanced source dataset achieved better performance. There seem to be many noisy neutral samples in the German dataset, by reducing its size, thus reducing the proportion of noisy neutral samples, seems to improve the ability to classify hateful comments. Table 5.2 shows the distribution of source samples for training. We split the target dataset into training, validation, and test set since we want to predict hate speech samples in the target language. During training, we evaluate the models on the validation set, and after training we evaluate it on the test set. The target dataset is split into 80% training and

Table 5.2: Source datasets summary.

Dataset	#non-hate	#hate	hate ratio	total
en_source	9,507	1,196	0.11	10,703
de_source	610	610	0.50	1,220
hr_source_12k	10,680	1,320	0.11	12,000

Table 5.3: Target datasets summary.

Dataset	#non-hate	#hate	hate ratio	total
en train	760	763	0.50	1,523
en val	197	194	0.50	391
en test	239	239	0.50	478
de train	391	375	0.49	766
de val	88	109	0.55	197
de test	131	126	0.49	257
hr_bal_train	960	971	0.50	1,931
hr_bal_val	227	221	0.49	448
hr_bal_test	313	308	0.50	621

20% test. Furthermore, the train set is split in the same way into training and validation set. To ensure comparability during our experiments, the validation and test set stay fixed and do not change. This means, that when we add target samples to the training, we always take samples from the train set. In Table 5.3, we report the distribution of target samples we use for training, validation, and test.

5.3 Models

To classify the samples, we have constructed two models. One is a modified version of the CNN structure, proposed by Kim [20]. The other tested model structure combines BiLSTM and CNN. The CNN is better suited for extracting features and the BiLSTM model is better suited for getting dependencies between tokens. With two separate models, we hope to see how classification performance and the influence of cross-lingual embeddings differ between model structures.

Since our models only accept a fixed length of input, we set the maximum sequence length to 100 tokens. If samples are shorter, we zero pad them, if they are longer, we shorten them to make them the appropriate length (this affects about 5% of all samples). The activation function for our hidden layers is ReLu. For output layer, we use the softmax activation function and for the loss function the categorical cross-entropy. Since we predict binary values, we could have also chosen sigmoid activation and binary cross-entropy loss functions. Our approach behaves the same way in binary classification with the added possibility of predicting more labels without changing model structures. We use AdaDelta as the optimiser. We have briefly experimented with Adam with worse performance.

To reduce overfitting during training, we employ early stopping. We monitor the validation loss and stop the training if the validation has not improved for 10 epochs. We consider the validation loss improved if the improvement between iterations is more than 0.001. The maximum number of epochs is 50.

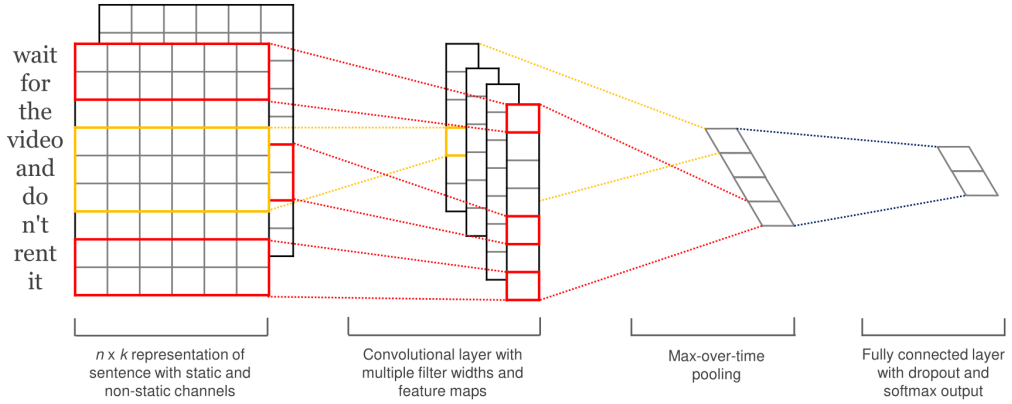

Figure 5.1: CNN model structure [20].

5.3.1 CNN

CNNs can capture features regardless of where they appear in the input. They are most often used in image recognition, but this method can also be used in text classification. In our case, this means that we can recognise patterns regardless of where they appear in the text and word order is less important.

Figure 5.1 shows the structure of our CNN model. The input of the model is the embedded sequence of the shape *maximum sequence length*$\times$*embedding size*. After the input layer, there is a convolutional layer with max-over-time pooling, which means that we are looking for local maximums in a 1D sequence of inputs. In this layer, there are multiple filters with different sizes and feature maps. We have used filter sizes 6, 5, 4, 3 and 2 $\times$ embedding size. This effectively means that we are looking for the most distinctive patterns in 6-grams to bi-grams. For regularisation, we use the dropout layer with 0.5 probability.

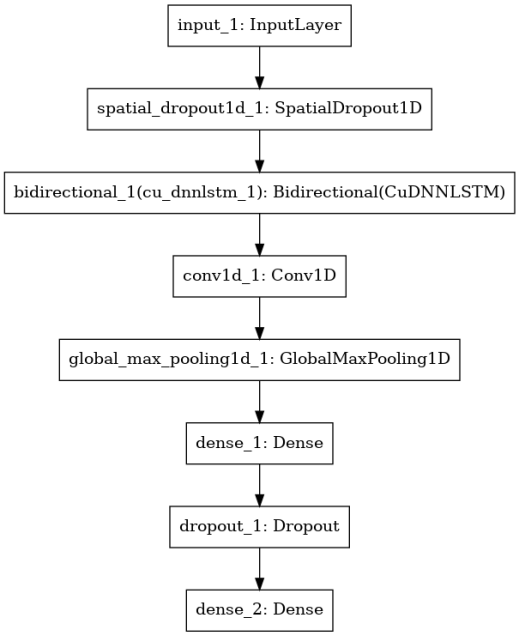

Figure 5.2: BiLSTM model structure.

5.3.2 BiLSTM-CNN

Since the meaning in text is often derived from the order of word appearance, LSTMs are often used in text classification tasks. LSTMs allow us to remember past input and make decisions based on that [16]. BiLSTMs learn by seeing input from both directions. This allows us to capture dependencies from both ends of text. Our model combines BiLSTM and CNN. We hope to capture dependencies between tokens with the BiLSTM and capture important local features with the convolutional layer.

Figure 5.2 shows the structure of our BiLSTM model. The input layer is the same as in the CNN model. After the input layer, we do a spatial dropout (dropping entire columns of the input matrix), since the embeddings are sparse. Following the dropout, we have a BiLSTM layer which feeds into a convolutional layer with max pooling. Before predictions, similar to our CNN model, we feed the input to a dropout layer with 0.1 probability.

5.4 Evaluation

We report classification results for all tested combinations. The chosen metrics are recall, precision and F1. Precision and recall are defined as

$$recall = \frac{\text{TP}}{\text{TP + FN}} \tag{5.1}$$

$$precision = \frac{\text{TP}}{\text{TP + FP}} \tag{5.2}$$

and for the F1 score we use sklearn's implementation of the weighted average F1 score [22].

We check how much added value samples from the source language have by adding target samples to training data. We incrementally increase the amount of target samples by 2.5% until we reach 20% of added samples. Then we test the model's performance with added 40%, 60%, and 100% of the target samples. We also check at the correlation between the scores and the amount of training samples used. Since we are focusing on models' ability to detect hate speech, we also test the training with only source hate speech samples.

Chapter 6

Results

In this chapter, we present the results for the six language combinations with the approach described in Chapter 5. Since we test 68 combinations per language pair, we present all results in Appendix A.1 for the well-aligned results, and in Appendix A.2 for the poorly aligned results. Here, we only show the summary of results. To better show the added value of source samples on classification results, we have prepared compact results. Compacted tables contain the best achieved results for each split (12 different target splits, from 0 to 20% added target samples with 2.5% increase per step, 40%, 60%, 80%, and 100% added target samples). ΔF1 values in compacted tables show the difference between the best and the second best result, with or without added target samples (depending on the best result), of the same model.

Analysing the results for the six language combinations, we noticed that we can group the results in two groups by the quality of cross-lingual embeddings. The combinations English-German and German-English achieve better results in terms of added value of source samples and have the best alignment scores (see Table 5.1). They represent the best case scenario. The remaining combinations have worse scores in terms of added value of source samples and alignment. In this chapter, we present the results in three sections, in the first the results for the well-aligned language combinations, followed by less well-aligned language combinations. In the third section, we

compare results of our approach with Multilingual BERT [13].

6.1 Well-aligned languages results

The language pairs English-German and German-English are our best aligned language combinations. In this section, we focus on the combination English-German since it represents the best case scenario for our approach.

The compacted results for English source and German target can be seen in Table 6.1. In all cases, we get an improvement of the F1 score when adding samples in the source language. When we have little (below 20%) of the target samples available, the added value of source samples is substantial. After that, it seems that we get diminishing returns, which seems intuitive since the number of source samples stays the same and only the number of target samples increases. The result for training without target data seems especially impressive, with the F1 score of 0.62, while the best achieved score with all source and all target samples is 0.71. Recall is a problem when training with no target samples, which is to be expected since no target samples were used in training. Even with as little as 20 added target samples, the recall increases from 0.45 to 0.64, which is a substantial gain. The BiLSTM-CNN profits from the source samples much more than the CNN model. The best result for every target split is achieved with added source samples.

To check if there is actually a relationship between results and training on aligned source samples, we calculated the correlation matrix between the metrics and the number of source and target samples. We calculated the correlation on the whole result Table A.1 in appendix. Table 6.2 shows the correlation between metrics and number of samples used. We can see that the number of source samples is strongly positively correlated with the metrics, especially precision and F1. One can say that the importance of source samples is comparable to the importance of target samples. Still, target samples are more important for classification, especially in recall the difference is substantial.

Table 6.1: Compact result for English source and German target with ΔF1 scores. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1), the difference between F1 scores of the same target split (ΔF1), and the model used.

	only_hate	#src	#tgt	r	p	F1	ΔF1	model
0	False	10703	766	0.6984	0.7097	0.7120	0.0121	BiLSTM
1	False	0	766	0.7460	0.6763	0.6999	-	BiLSTM
2	False	10703	460	0.7619	0.6809	0.7075	0.0526	BiLSTM
3	False	0	460	0.8810	0.6133	0.6549	-	BiLSTM
4	False	10703	307	0.8492	0.6045	0.6410	0.0561	BiLSTM
5	False	0	307	0.7222	0.5652	0.5849	-	BiLSTM
6	False	10703	154	0.6984	0.6377	0.6572	0.0786	CNN
7	False	0	154	0.6349	0.5634	0.5786	-	CNN
8	False	10703	135	0.6667	0.6774	0.6809	0.1385	BiLSTM
9	False	0	135	0.6190	0.5306	0.5424	-	BiLSTM
10	False	10703	115	0.6905	0.6591	0.6731	0.1261	BiLSTM
11	False	0	115	0.6111	0.5347	0.5470	-	BiLSTM
12	False	10703	96	0.7302	0.6619	0.6843	0.1419	BiLSTM
13	False	0	96	0.6190	0.5306	0.5424	-	BiLSTM
14	False	10703	77	0.7937	0.6098	0.6430	0.1027	BiLSTM
15	False	0	77	0.5794	0.5290	0.5403	-	BiLSTM
16	False	10703	58	0.6905	0.6304	0.6494	0.107	BiLSTM
17	False	0	58	0.6190	0.5306	0.5424	-	BiLSTM
18	False	10703	39	0.5476	0.6330	0.6204	0.0747	CNN
19	False	0	39	0.4286	0.5567	0.5457	-	CNN
20	False	10703	20	0.6429	0.6750	0.6728	0.1304	BiLSTM
21	False	0	20	0.6190	0.5306	0.5424	-	BiLSTM
22	False	10703	0	0.4524	0.6951	0.6219	-	BiLSTM

Table 6.2: Correlation matrix for metrics of English source and German target.

	r	p	F1
#src	0.165068	0.562275	0.512695
#tgt	0.453470	0.529261	0.626205

We cannot directly compare our results with the results of the authors of the German dataset [8] since our training and test splits are different and they also report results for two out of the three subsets of the whole dataset, while we use the whole dataset. Assuming that the average results of the two datasets represent the whole dataset, they get an estimated recall of 0.58, precision of 0.75 and F1 score of 0.65. Generally, we achieve worse precision, improve on recall and improve the F1 score.

The German-English combination is equally well-aligned to English-German, but the added value of source samples is smaller. All the results can be found in Appendix A.1. In Table A.2, we report the results for German as the source language and English as the target language. We can observe increased score with added target samples, as expected. Similarly to before, we observe that the added value of source samples is larger when we have little to no target samples available. Adding source samples improves the results and when it does not, the ΔF1 is 0.015 or below. This could be due to the fact that the English dataset is sentence-level labelled and the German is document-level labelled. This makes the German samples noisier. For example, a sample can have multiple sentences and only one of them is hate speech. In document-level labelling, we label the whole document as hate speech even though not all sentences are hate speech. Again, results of training without target samples are acceptable. Results are comparable to the results when training with 115 target samples, with worse recall. The results for training with no target samples seems especially impressive, with the F1 score of 0.69 (recall of 0.63 and precision of 0.71), which is 0.11 lower than the best result.

Table A.3 shows the correlation between results and the number of samples used, calculated on the whole result Table A.4 in the appendix. We observe that source samples are positively correlated with precision and F1 and negatively correlated with recall. This seems to be due (as discussed before) to the document-level labelling of the German dataset. Target samples in this language combination are more valuable than source samples.

We have used the same train and test dataset as the authors of the original

study [12]. Still, we cannot directly compare the results since the metrics used are different. Our best model achieves a recall of 0.83 and accuracy of 0.78 (with less target samples), which improves upon the author's recall of 0.71 and accuracy of 0.73 (see Table 3.3).

We can conclude that if languages are well-aligned, source samples add value to classification results. Especially with a low number of target samples, the added value is substantial. While increasing the amount of target samples in training, we observe diminishing returns from source samples in regards to the observed scores. Even if there are no available target samples, training only on source samples achieves acceptable performance. The recall is problematic in this case, but even a small number of target samples substantially increases the recall.

6.2 Poorly aligned languages results

In this section, we discuss results when the alignment of source and target languages is poor, i.e. all combinations with Croatian. This seems to be due to Croatian not being similar to neither German nor English. We take a closer look at the combination Croatian-German, as the case of bad alignment, and Croatian-English, as the worst case of bad alignment. The other two combinations offer no new insights, so we do not comment them. Still, the results for German-Croatian and English-Croatian are available in the appendix.

In Table 6.3, there are results for Croatian source and German target, our best case of poor alignment. Training with no target samples is not a good option for this language combination, considering it is practically unable to recognise any hate speech (recall of 0.03). The only substantial added value of source samples in this combination is when we add 20 samples, where the ΔF1 is 0.22, achieving the F1 score of 0.57. This is still low compared to our best F1 score of 0.75. Contrary to our observations in the previous section, the added value of source samples is consistent through all target splits. This

points that there is no pattern for added value of source samples when the alignment is poor. A negative outlier is training with all source samples and 460 target samples, where adding source samples decreases the F1 score by 0.07. This bad performance is mainly due to the badly aligned source space, which means that training the model with source samples mostly adds noise, making hate speech recognition in the target space harder.

Even though the alignment is poor, the source samples add more value than expected to F1 score, with a correlation of 0.11 (see Figure 6.4). Source samples add little to precision (correlation of 0.04) and are negatively correlated with recall (correlation of –0.12). As previously mentioned, recall seems to be affected by the added noise of the poorly aligned source samples.

Our worst case example of bad alignment is the language combination Croatian-English. In the appendix Table A.6, there are compacted results for Croatian source and English target. There is no substantial performance increase when we add source samples. The source samples do not have an impact in any of the target splits, the biggest gain being 0.04, and the biggest loss being 0.04. Because of that, we assume that the hate speech overlap between the embedded spaces is very small.

Table 6.5 shows the correlation between results and number of samples. The number of source samples is negatively correlated with all the scores and especially recall with a value of –0.27. There is no influence in precision, and correlation with F1 score is negative with the value of –0.11. Training with no target samples skew the correlation because of the poor performance. If we remove those, we get a positive correlation of 0.09 with the F1 score, 0.20 with precision, but recall remains negatively correlated at –0.14. The added value of source samples between models differs substantially. Ignoring trained with no target samples, the CNN model has a negative correlation with the F1 score of –0.10, while the BiLSTM-CNN model seems to profit from source samples with correlation of 0.28 to the F1 score. All the best scores of the BiLSTM-CNN model were achieved with added source samples. This suggests that even with poor alignment we can extract dependencies from

Table 6.3: Compact result table for Croatian source and German target with ΔF1 scores. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1) and the difference between F1 scores of the same target split (ΔF1), and the model used.

	only_hate	#src	#tgt	r	p	F1	ΔF1	model
0	False	0	766	0.7937	0.7299	0.7546	-	CNN
1	True	1320	766	0.7698	0.7132	0.7352	-0.0194	CNN
2	False	0	460	0.7302	0.7360	0.7393	-	CNN
3	False	12000	460	0.7857	0.6346	0.6694	-0.0699	CNN
4	False	12000	307	0.6984	0.6331	0.6532	0.0094	CNN
5	False	0	307	0.7857	0.6111	0.6438	-	CNN
6	False	12000	154	0.7381	0.6458	0.6720	0.0119	CNN
7	False	0	154	0.7302	0.6345	0.6601	-	CNN
8	False	0	135	0.7222	0.6454	0.6685	-	CNN
9	False	12000	135	0.5873	0.6727	0.6558	-0.0127	CNN
10	False	12000	115	0.5476	0.6635	0.6388	0.0196	CNN
11	False	0	115	0.7698	0.5915	0.6192	-	CNN
12	False	12000	96	0.6032	0.6179	0.6224	0.0119	CNN
13	False	0	96	0.5794	0.6083	0.6105	-	CNN
14	False	12000	77	0.5952	0.5906	0.5992	0.0267	CNN
15	False	0	77	0.4206	0.6092	0.5725	-	CNN
16	False	12000	58	0.5873	0.6167	0.6183	0.0081	CNN
17	False	0	58	0.6587	0.5929	0.6102	-	CNN
18	False	12000	39	0.5238	0.6055	0.5969	0.0523	CNN
19	False	0	39	0.8016	0.5401	0.5446	-	CNN
20	False	12000	20	0.4444	0.5895	0.5686	0.2244	BiLSTM
21	False	0	20	0.0000	0.0000	0.3442	-	BiLSTM
22	False	12000	0	0.0317	0.4444	0.3680	-	CNN

Table 6.4: Correlation matrix for metrics of Croatian source and German target, our best case example of a bad alignment.

	r	p	F1
#src	-0.119145	0.044716	0.111110
#tgt	0.261762	0.385075	0.653312

Table 6.5: Correlation matrix for metrics of Croatian source and English target, our worst case example of a bad alignment. The results with removed instances, trained with no target samples, are marked with an asterisk (*).

	r	p	F1
#src	-0.273637	-0.003104	-0.116128
#tgt	0.297047	0.384704	0.575193
#src*	-0.146265	0.204576	0.092097
#tgt*	0.320556	0.599150	0.707474

the source language, which help predict hate speech in the target language.

German-Croatian and English-Croatian are poorly aligned language pairs, where adding source samples has little to no influence on results. The biggest F1 score gain in these combinations is 0.06, while the biggest loss is –0.04. Other gains hover around +/–0.02, as visible in Tables A.8 and A.11 in the appendix.

We notice that Croatian as a source language offers two extremes, German as the target language is the best case, and English as the target is the worst case. As previously noted, this seems to be due to Croatian and German being labelled at a document level, and English being labelled at sentence level. Considering the quality of alignments, we assumed that the best between poorly aligned language combinations would get most added value from source samples. This does not seem to be the case. Croatian-

English is best aligned, and yet the performance is worst, while the second worst alignment performs best in the group of the poorly aligned languages. This seems to confirm what Glavas et al. [15] are claiming, BLI performance does not necessarily correlate to performance in downstream tasks.

Generally, we find that if the languages are not well-aligned, the added value of source samples is negligible. Even in our best case, where the F1 score was positively correlated, the added value of source samples is negligible (at most 0.02) as soon as we add more than 77 samples. In practice, none of the poorly aligned models trained only on source samples are usable.

6.3 Comparison with Multilingual BERT

Because of the poor performance of combinations with Croatian, we checked if this is due to poor alignments. To that aim, we employed Multilingual BERT [13], which is currently the state-of-the-art method for cross-lingual language representation. In domains where data is scarce, BERT improves performance. BERT models are trained on large general corpora, so fine-tuning them on domain specific data drastically improves the performance. In this section, we compare results achieved with RCSLS alignments and Multilingual BERT.

We use Multilingual BERT language model and fine-tune it using textual data from source datasets in English, German, and Croatian. The Language model fine-tuning script from PyTorch-Transformers [1] is used for fine-tuning with default parameters and three epochs. The script fine-tunes all hidden layers. We feed the BERT output to the classifiers, introduced in Section 5.3.

During preliminary testing, we noticed that models trained on the full source datasets get stuck in local minima. This could be solved with hyper-parameter optimisation. However, we could reduce this issue using a smaller balanced target train sets as train sets in the source language instead of the bigger imbalanced source datasets. All results are achieved with target train

Table 6.6: Comparison of RCSLS alignments and Multilingual BERT. Training was done with source and target samples (all), only target samples (tgt) and only source samples (src). We report recall (r), precision (p), F1 score (f) for the best result (out of both models) for the three splits. The best result of each column is bolded.

	en-de			en-hr			de-en			de-hr			hr-en			hr-de		
	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f	r	p	f
BERT all	**0.75**	0.67	0.70	**0.81**	0.69	**0.75**	0.87	0.72	0.79	**0.76**	0.71	**0.73**	0.84	0.72	**0.78**	0.78	0.62	0.69
RCSLS all	0.70	**0.71**	0.71	0.71	0.69	0.70	0.80	0.75	0.77	0.71	0.72	0.72	0.81	0.75	0.77	0.77	0.71	0.74
BERT tgt	0.72	**0.71**	**0.72**	0.74	**0.74**	0.74	**0.88**	0.72	0.79	0.71	**0.75**	**0.73**	0.85	0.73	**0.78**	0.73	0.69	0.71
RCSLS tgt	**0.75**	0.68	0.70	0.65	0.72	0.70	0.75	**0.84**	**0.80**	0.72	0.70	0.71	**0.90**	0.73	**0.78**	**0.79**	**0.73**	**0.76**
BERT src	0.65	0.64	0.64	0.49	0.58	0.53	0.63	0.67	0.65	0.67	0.61	0.64	0.77	0.64	0.70	0.71	0.53	0.61
RCSLS src	0.45	0.70	0.62	0.24	0.64	0.51	0.63	0.71	0.69	0.19	0.59	0.47	0.11	**0.84**	0.44	0.03	0.44	0.37

sets in the source language.

In Table 6.6, we present the best results with Multilingual BERT and RCSLS alignments. On average, using all samples or only target samples yields similar performance, so we skip further discussion. The most extreme cases are Croatian-German, where RCSLS achieves 0.05 higher F1 score, and English-Croatian, where BERT achieves 0.05 higher F1 score. When comparing results trained with only source samples, BERT outperforms RCSLS in all language pairs except German-English, where RCSLS achieves a 0.04 higher F1 score. F1 score improvements in German-Croatian (+0.17), Croatian-English (+0.16), and Croatian-German (+0.24) are substantial. The combinations English-German and Croatian-English have the smallest F1 score delta (0.08) to the best result, which is surprising for Croatian-English since the languages are not similar. In practice, BERT performance without target samples in English-German, German-Croatian, and Croatian-English seems acceptable (F1 score delta to the best result are lower than 0.10). This seems to confirm that the poor performance of combinations with Croatian was caused by the poor RCSLS alignments.

Table A.14 in the appendix presents all tested BERT combinations. The BiLSTM-CNN significantly outperforms the CNN model, which stucks in

local minima, so training often fails after the first epoch. This seems to be due to the used hyperparameters.

We also fine-tuned 11 hidden layers of the BERT model and added a softmax layer for classification. During testing on a small subset of language combinations, this approach yielded slightly worse results, so we skip further discussion.

Our experiments confirm that Multilingual BERT shows improved performance in cross-lingual transfer and should therefore be the preferred method for cross-lingual embeddings. Considering that we have used less training samples than in the RCSLS approach and significantly improved results using only source samples, we can assume that with classifier tuning the scores could further improve.

Chapter 7

Conclusion

The goal of this work was to develop an approach that uses cross-lingual embeddings to solve the problem of hate speech detection in low-resource languages. We have chosen the RCSLS method for alignment of fastText vectors and developed two models, CNN and BiLSTM-CNN, for classification. We use the BERT model to compare it to the RCSLS method. We evaluated the approach on six language combinations. Simulating a low-resource language, we trained the models on larger source datasets (at most 12,000 samples) and tested on small target datasets (at most 600 samples). The performance metrics used were recall, precision and F1 score.

Our initial assumption was that cross-lingual embeddings will transfer some information from a source to target language. Due to the noisy domain, we did not know how much that will affect the impact of provided source data. The most important findings are:

1. **Best case:** If the languages are well-aligned, the source samples are substantially positively correlated with the performance metrics. Especially in the combination English-German, we observe that the source samples are almost as important as the target samples. If we have no samples available in the target language, we still achieve acceptable performance. The F1 score difference between the best achieved result with all target samples and with no target samples is around 0.10 for

both well-aligned language combinations. The added value of source samples is the largest with very little target samples; with increasing number of added target samples we get diminishing return of source samples.

2. **Worst case:** When the languages are poorly aligned the number of source samples are, in the best case, positively correlated with F1 score correlation of 0.11. In the worst case, they are equally negatively correlated. In tested language combinations, we can expect a negative F1 score correlation when the alignment is poor, since three out of four combinations show negative correlation. When no target samples are used in training, the resulting model is not usable. Even when we start adding target samples, the added value of source samples is mostly negligible.

Even though we cannot directly compare performance scores due to different testing circumstances, the models proposed seem to work better (under similar circumstances) compared to models proposed by the authors of the English [12] and German dataset [8].

The metrics we use to evaluate the quality of cross-lingual embeddings measure BLI performance, e.g., nearest neighbour and CSLS. BLI performance, however, is not necessarily a good indicator of performance on downstream tasks like hate speech detection [15]. In our poorly aligned language pairs, the second worst aligned language pair achieved the best result, and was the only positively correlated poorly aligned language pair. The second best was the most negatively correlated. This suggests that BLI performance is indeed not a good indicator of downstream task performance, at least for poorly aligned language pairs.

To check if the poor performance of language combinations with Croatian is caused by the poor alignments, we used Multilingual BERT as the cross-lingual language model. We found that trained without target samples, Multilingual BERT significantly improves performance on language pairs where alignment is poor. In other cases, performance is comparable.

We consider the goals of the thesis reached. In the case of well-aligned languages, the proposed approach works well. For languages that are further apart, and intuitively hard to align, our approach does not work well.

7.1 Limitations and future work

We discuss the limitations of our work and possible research directions for the future.

We have not performed any hyperparameter optimisation, nor have we been trying to find the optimal network architecture. To perform hyperparameter tuning, we could use sklearn's "GridSearchCV" [22]. Performing this on all the tested combinations would be very time intensive, so we could test the procedure on a small subset of instances. Tuning a model trained with only source samples, and a model trained with source samples and all target samples, seems sensible and would give the most added value. Another limitation of our models is that we likely overfit the data, and even though we have tried to minimise overfitting by adding dropout layers and early stopping, the number of trainable parameters is much larger than the number of observed samples. This affects robustness of our models, which we can see in some results being dependant on the samples chosen in data splits. E.g. we add more target samples to training, but the performance drops, even though the opposite is expected. This problem is apparent when we were testing Multilingual BERT. Our models were prone to getting stuck in local minima.

While we have tried to find hate speech datasets as similar as possible to each other, the datasets still differ when it comes to the type of hate speech, e.g., white supremacy forum v.s. Pegida Facebook group. The English and German datasets are targeted towards a specific type of hate speech, while the Croatian dataset seems to be less targeted.

Further limitation is that we have used pre-trained word-embeddings with a dictionary that is non-specific to our problem. We do not specifically con-

sider slang words and synonyms which may appear in our dataset and not in the dictionary. Using a hate lexicon could be a possible solution. Training an embedding specific to our problem would probably increase the model's performance. In a similar vein, we could have solved the problem of out-of-vocabulary (OOV) words by using fastText's feature that can build an embedding of a OOV word by splitting it into a bag of n-grams and then summing the representations of those as the representation of the OOV word. Since we cannot align fastText models, we could get the vector representations of all OOV words in the training dataset and add them to the fastText word vector embedding which we can align.

Though we have achieved relatively good classification results, we did not aim to maximise them. An approach that would most likely improve upon our best models, would need three changes. Our preprocessing is basic, and does not consider information gained from e.g. hashtags and misspellings. Ekphrasis [5] seems to be a good choice for social media content preprocessing. Multilingual BERT [13] may be the best choice as contextualised embedding, since it is currently widely used as the state-of-the-art language model. For classification, the skipped CNN proposed by Zhang and Luo [33] is, to the best of our knowledge, the best model architecture for hate speech.

Since hate speech samples are rare, it makes sense to acquire more samples. One way to do it is to use text-augmentation techniques, e.g., to randomly switch order of tokens in hate speech samples to create more samples. We assume that the CNN model would benefit from such an approach. Another approach is to switch chosen words with their synonyms and create new hate speech samples in that way. This would expand the area hate speech occupies in the embedded space, making classification easier.

One approach to hate speech detection is often dependant on context, as seen by the hate speech sample in the Croatian dataset (see Table 3.6). It would make sense to consider previous comments to classify hate speech, e.g., if a comment is hate speech and it has a response which supports it, even though the response by itself is not considered hate speech, it should

be considered as such.

To make samples in a source language a viable option as a replacement for samples in the target language, further improvements are needed in the field of cross-lingual alignment. Especially for languages that are not close, alignments need further improvement. There is also the question of how to evaluate cross-lingual embeddings for downstream tasks such as ours. An experiment with major cross-lingual embedding methods on hate speech datasets would be beneficial.

We should note that this work ignores the LASER toolkit introduced by Artetxe and Schwenk [3], which is another state-of-the-art cross-lingual mapping model. It uses a single language-agnostic BiLSTM encoder for 93 languages, which was trained on publicly available parallel corpora and applied to different downstream tasks. All languages are jointly embedded in a shared space, in contrast to most other works which usually separately consider English and foreign alignments [3]. It is sensible to test it in the domain of hate speech detection.

Appendix A

Complete results

In the appendix, we present all tables for the six language combinations tested, related to Chapter 6. We tested 68 data split combinations. We have split the results in two sections, in the first, we present the results for the well-aligned language combinations, and in the second section, we present the results for the poorly aligned language combinations. The tables found in the appendix are the compacted tables, full result tables and correlation tables.

A.1 Well-aligned results

In this section, we present complete results for well-aligned language pairs English-German and German-English. Below you can find the following tables:

- Full result table for English-German in Table A.1.
- Tables for the language pair German-English: the compacted result Table A.2, correlation matrix in Table A.3, and the full in Table A.4.

Table A.1: Results for English source and German target. CNN on the left and BiLSTM on the right side. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1).

	CNN							BiLSTM					
	only_hate	#src	#tgt	r	p	F1		only_hate	#src	#tgt	r	p	F1
0	False	0	766	0.6508	0.7009	0.6920	0	False	10703	766	0.6984	0.7097	**0.7120**
1	False	10703	766	0.6825	0.6719	0.6810	1	True	1196	766	0.7540	0.6786	0.7037
2	True	1196	766	0.7143	0.6522	0.6727	2	False	0	766	0.7460	0.6763	0.6999
3	False	0	460	0.8492	0.6446	0.6899	3	False	10703	460	0.7619	0.6809	**0.7075**
4	False	10703	460	0.8413	0.6199	0.6600	4	False	0	460	0.8810	0.6133	0.6549
5	True	1196	460	0.7698	0.6178	0.6494	5	True	1196	460	0.6825	0.5890	0.6091
6	False	10703	307	0.5397	0.6667	0.6382	6	False	10703	307	0.8492	0.6045	**0.6410**
7	False	0	307	0.6825	0.6099	0.6295	7	False	0	307	0.7222	0.5652	0.5849
8	True	1196	307	0.5952	0.5906	0.5992	8	True	1196	307	0.6587	0.5497	0.5648
9	False	10703	154	0.6984	0.6377	**0.6572**	9	False	10703	154	0.6667	0.6222	0.6379
10	True	1196	154	0.6111	0.5833	0.5953	10	True	1196	154	0.6984	0.5789	0.5998
11	False	0	154	0.6349	0.5634	0.5786	11	False	0	154	0.6190	0.5306	0.5424
12	False	10703	135	0.6984	0.6069	0.6289	12	False	10703	135	0.6667	0.6774	**0.6809**
13	True	1196	135	0.5635	0.6017	0.6025	13	True	1196	135	0.6190	0.5306	0.5424
14	False	0	135	0.5952	0.5556	0.5679	14	False	0	135	0.6190	0.5306	0.5424
15	False	10703	115	0.4683	0.6413	0.6029	15	False	10703	115	0.6905	0.6591	**0.6731**
16	True	1196	115	0.5556	0.5983	0.5985	16	False	0	115	0.6111	0.5347	0.5470
17	False	0	115	0.5714	0.5714	0.5798	17	True	1196	115	0.6190	0.5306	0.5424
18	False	10703	96	0.6190	0.6240	0.6303	18	False	10703	96	0.7302	0.6619	**0.6843**
19	True	1196	96	0.5952	0.5639	0.5758	19	True	1196	96	0.6190	0.5306	0.5424
20	False	0	96	0.5556	0.5691	0.5757	20	False	0	96	0.6190	0.5306	0.5424
21	False	0	77	0.5159	0.5752	0.5744	21	False	10703	77	0.7937	0.6098	**0.6430**
22	False	10703	77	0.4206	0.6023	0.5691	22	True	1196	77	0.6190	0.5306	0.5424
23	True	1196	77	0.6349	0.5369	0.5497	23	False	0	77	0.5794	0.5290	0.5403
24	False	10703	58	0.5556	0.6195	0.6134	24	False	10703	58	0.6905	0.6304	**0.6494**
25	True	1196	58	0.6746	0.5667	0.5847	25	True	1196	58	0.6190	0.5306	0.5424
26	False	0	58	0.5317	0.5776	0.5788	26	False	0	58	0.6190	0.5306	0.5424
27	False	10703	39	0.5476	0.6330	**0.6204**	27	False	10703	39	0.8810	0.5812	0.6073
28	True	1196	39	0.6111	0.5620	0.5755	28	True	1196	39	0.5873	0.5362	0.5481
29	False	0	39	0.4286	0.5567	0.5457	29	False	0	39	0.6190	0.5306	0.5424
30	False	10703	20	0.5714	0.6050	0.6065	30	False	10703	20	0.6429	0.6750	**0.6728**
31	True	1196	20	0.4683	0.5221	0.5275	31	True	1196	20	0.6190	0.5306	0.5424
32	False	0	20	0.0476	0.3158	0.3666	32	False	0	20	0.6190	0.5306	0.5424
33	False	10703	0	0.6349	0.5755	0.5908	33	False	10703	0	0.4524	0.6951	**0.6219**

Table A.2: Compact result table for German source and English target with ΔF1 scores. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1), and model used.

	only_hate	#src	#tgt	r	p	F1	ΔF1	model
0	False	0	1523	0.7448	0.8357	0.7986	-	CNN
1	False	1220	1523	0.7950	0.7540	0.7676	-0.031	CNN
2	True	610	914	0.7866	0.7769	0.7803	0.0291	CNN
3	False	0	914	0.8828	0.7033	0.7512	-	CNN
4	True	610	610	0.8243	0.7695	0.7884	0.0353	CNN
5	False	0	610	0.7657	0.7469	0.7531	-	CNN
6	False	0	305	0.8410	0.7309	0.7644	-	CNN
7	False	1220	305	0.6862	0.7961	0.7541	-0.0103	CNN
8	False	0	267	0.7699	0.7449	0.7531	-	CNN
9	True	610	267	0.8117	0.7106	0.7393	-0.0138	CNN
10	False	0	229	0.8075	0.7338	0.7567	-	CNN
11	False	1220	229	0.7908	0.7326	0.7507	-0.006	CNN
12	False	1220	191	0.6862	0.7664	0.7378	0.0019	CNN
13	False	0	191	0.7782	0.7181	0.7359	-	CNN
14	False	1220	153	0.6862	0.7558	0.7316	0.0015	CNN
15	False	0	153	0.7322	0.7292	0.7301	-	CNN
16	False	1220	115	0.8201	0.6853	0.7190	0.1948	BiLSTM
17	False	0	115	0.9205	0.5473	0.5242	-	BiLSTM
18	False	1220	77	0.6695	0.7442	0.7190	0.0809	CNN
19	False	0	77	0.7490	0.6172	0.6381	-	CNN
20	False	1220	39	0.6569	0.7202	0.7003	0.0937	CNN
21	False	0	39	0.7155	0.5917	0.6066	-	CNN
22	False	1220	0	0.6318	0.7123	0.6873	-	CNN

Table A.3: Correlation matrix for metrics of German source and English target

	r	p	F1
#src	-0.051604	0.316607	0.247838
#tgt	0.348551	0.468938	0.536512

Table A.4: Results for German source and English target. CNN on the left and BiLSTM on the right side. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1).

	CNN						BiLSTM					
	only_hate	#src	#tgt	r	p	F1	only_hate	#src	#tgt	r	p	F1
0	False	0	1523	0.7448	0.8357	**0.7986**	False	1220	1523	0.8201	0.7368	0.7628
1	False	1220	1523	0.7950	0.7540	0.7676	False	0	1523	0.8745	0.7133	0.7584
2	True	610	1523	0.8703	0.7051	0.7497	True	610	1523	0.8368	0.7168	0.7514
3	True	610	914	0.7866	0.7769	**0.7803**	False	1220	914	0.7490	0.7490	0.7490
4	False	1220	914	0.7866	0.7611	0.7698	False	0	914	0.8536	0.6962	0.7372
5	False	0	914	0.8828	0.7033	0.7512	True	610	914	0.8075	0.7044	0.7329
6	True	610	610	0.8243	0.7695	**0.7884**	True	610	610	0.8703	0.6775	0.7224
7	False	0	610	0.7657	0.7469	0.7531	False	1220	610	0.8536	0.6800	0.7214
8	False	1220	610	0.7406	0.7597	0.7531	False	0	610	0.8368	0.6826	0.7203
9	False	0	305	0.8410	0.7309	**0.7644**	False	1220	305	0.7782	0.7019	0.7230
10	False	1220	305	0.6862	0.7961	0.7541	False	0	305	0.6569	0.7009	0.6880
11	True	610	305	0.8285	0.6996	0.7341	True	610	305	0.7573	0.6558	0.6780
12	False	0	267	0.7699	0.7449	**0.7531**	False	1220	267	0.7238	0.7119	0.7155
13	True	610	267	0.8117	0.7106	0.7393	True	610	267	0.5816	0.7473	0.6886
14	False	1220	267	0.7699	0.7160	0.7318	False	0	267	0.7071	0.6550	0.6668
15	False	0	229	0.8075	0.7338	**0.7567**	False	1220	229	0.7197	0.7382	0.7322
16	False	1220	229	0.7908	0.7326	0.7507	True	610	229	0.8619	0.6059	0.6343
17	True	610	229	0.8075	0.6918	0.7219	False	0	229	0.8828	0.5687	0.5742
18	False	1220	191	0.6862	0.7664	**0.7378**	False	1220	191	0.7908	0.6750	0.7028
19	False	0	191	0.7782	0.7181	0.7359	False	0	191	0.8410	0.6128	0.6424
20	True	610	191	0.8033	0.6621	0.6932	True	610	191	0.6527	0.6265	0.6316
21	False	1220	153	0.6862	0.7558	**0.7316**	False	1220	153	0.7197	0.6935	0.7007
22	False	0	153	0.7322	0.7292	0.7301	True	610	153	0.7908	0.6117	0.6366
23	True	610	153	0.6862	0.7009	0.6966	False	0	153	0.7573	0.5801	0.5952
24	False	1220	115	0.7741	0.6981	0.7188	False	1220	115	0.8201	0.6853	**0.7190**
25	False	0	115	0.7908	0.6585	0.6872	True	610	115	0.8452	0.5788	0.5935
26	True	610	115	0.7908	0.6540	0.6827	False	0	115	0.9205	0.5473	0.5242
27	False	1220	77	0.6695	0.7442	**0.7190**	False	1220	77	0.8033	0.6214	0.6494
28	True	610	77	0.6360	0.6609	0.6547	False	0	77	0.7113	0.6204	0.6361
29	False	0	77	0.7490	0.6172	0.6381	True	610	77	0.7741	0.6106	0.6336
30	False	1220	39	0.6569	0.7202	**0.7003**	False	1220	39	0.5941	0.7358	0.6875
31	False	0	39	0.7155	0.5917	0.6066	True	610	39	0.3431	0.5816	0.5283
32	True	610	39	0.2762	0.5546	0.4954	False	0	39	0.0879	0.5676	0.4040
33	False	1220	0	0.6318	0.7123	**0.6873**	False	1220	0	0.7699	0.6133	0.6363

A.2 Poorly aligned results

In this chapter of the appendix, we present the full results for the poorly aligned language pairs. Below you can find the following tables:

- Full result tables for Croatian-German in Table A.5.
- Tables for the language pair Croatian-English, the compacted result table in Table A.6, and the full result table in Table A.7.
- Tables for the language pair English-Croatian, the compacted result table in Table A.8, correlation matrix in Table A.9, and the full result table in Table A.10.
- Tables for the language pair German-Croatian, the compacted result table in Table A.11, correlation matrix in Table A.12, and the full result table in Table A.13.

A.3 Multilingual BERT results

In this section, we present the results for Multilingual BERT as an embedding in Table A.14. We tested three source and target data split combinations for every language pair: training on all source samples, training on all target samples, and training on all target and source samples.

Table A.5: Results for Croatian source and German target. CNN on the left and BiLSTM on the right side. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1).

	CNN							BiLSTM					
	only_hate	#src	#tgt	r	p	F1		only_hate	#src	#tgt	r	p	F1
0	False	0	766	0.7937	0.7299	**0.7546**	0	False	0	766	0.7063	0.7417	0.7351
1	True	1320	766	0.7698	0.7132	0.7352	1	False	12000	766	0.7381	0.7099	0.7237
2	False	12000	766	0.8016	0.6871	0.7223	2	True	1320	766	0.6825	0.7167	0.7118
3	False	0	460	0.7302	0.7360	**0.7393**	3	True	1320	460	0.6905	0.6591	0.6731
4	False	12000	460	0.7857	0.6346	0.6694	4	False	12000	460	0.7778	0.6049	0.6359
5	True	1320	460	0.7302	0.6389	0.6642	5	False	0	460	0.7937	0.5917	0.6210
6	False	12000	307	0.6984	0.6331	**0.6532**	6	False	12000	307	0.6905	0.6000	0.6210
7	False	0	307	0.7857	0.6111	0.6438	7	False	0	307	0.6270	0.5766	0.5910
8	True	1320	307	0.7460	0.6026	0.6300	8	True	1320	307	0.7143	0.5488	0.5637
9	False	12000	154	0.7381	0.6458	**0.6720**	9	False	0	154	0.7460	0.6065	0.6343
10	False	0	154	0.7302	0.6345	0.6601	10	False	12000	154	0.6270	0.6220	0.6304
11	True	1320	154	0.7619	0.6038	0.6330	11	True	1320	154	0.0000	0.0000	0.3442
12	False	0	135	0.7222	0.6454	**0.6685**	12	False	0	135	0.6984	0.6286	0.6491
13	False	12000	135	0.5873	0.6727	0.6558	13	True	1320	135	0.7302	0.6013	0.6270
14	True	1320	135	0.6746	0.5556	0.5720	14	False	12000	135	0.7222	0.5549	0.5717
15	False	12000	115	0.5476	0.6635	**0.6388**	15	False	0	115	0.7937	0.5495	0.5609
16	False	0	115	0.7698	0.5915	0.6192	16	False	12000	115	0.7778	0.5444	0.5542
17	True	1320	115	0.7540	0.5556	0.5720	17	True	1320	115	0.0000	0.0000	0.3442
18	False	12000	96	0.6032	0.6179	**0.6224**	18	False	12000	96	0.6905	0.5800	0.6004
19	False	0	96	0.5794	0.6083	0.6105	19	True	1320	96	0.6349	0.5714	0.5868
20	True	1320	96	0.7143	0.5233	0.5274	20	False	0	96	0.8016	0.5459	0.5545
21	False	12000	77	0.5952	0.5906	**0.5992**	21	False	12000	77	0.7619	0.5486	0.5616
22	False	0	77	0.4206	0.6092	0.5725	22	False	0	77	0.8968	0.5305	0.5074
23	True	1320	77	0.5476	0.5111	0.5211	23	True	1320	77	0.8333	0.5000	0.4553
24	False	12000	58	0.5873	0.6167	**0.6183**	24	False	12000	58	0.7302	0.5644	0.5840
25	False	0	58	0.6587	0.5929	0.6102	25	False	0	58	0.8016	0.5611	0.5785
26	True	1320	58	0.6190	0.5065	0.5128	26	True	1320	58	1.0000	0.4903	0.3226
27	False	12000	39	0.5238	0.6055	**0.5969**	27	False	12000	39	0.8333	0.5072	0.4725
28	False	0	39	0.8016	0.5401	0.5446	28	False	0	39	0.8413	0.4977	0.4464
29	True	1320	39	0.7302	0.4946	0.4759	29	True	1320	39	1.0000	0.4903	0.3226
30	False	12000	20	0.3968	0.6098	**0.5655**	30	False	12000	20	0.4444	0.5895	0.5686
31	True	1320	20	0.7778	0.4949	0.4629	31	True	1320	20	0.5556	0.5036	0.5129
32	False	0	20	0.0476	1.0000	0.3942	32	False	0	20	0.0000	0.0000	0.3442
33	False	12000	0	0.0317	0.4444	**0.3680**	33	False	12000	0	0.0000	0.0000	0.3442

Table A.6: Compact result table for Croatian source and English target with ΔF1 scores. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1), and model used.

	only_hate	#src	#tgt	r	p	F1	ΔF1	model
0	False	0	1523	0.8996	0.7264	0.7772	-	CNN
1	True	1320	1523	0.8117	0.7549	0.7737	-0.0035	CNN
2	False	0	914	0.8285	0.7615	0.7841	-	CNN
3	False	12000	914	0.8326	0.7481	0.7754	-0.0087	CNN
4	True	1320	610	0.8033	0.7471	0.7654	0.0031	CNN
5	False	0	610	0.8368	0.7299	0.7623	-	CNN
6	False	0	305	0.7448	0.7876	0.7718	-	CNN
7	True	1320	305	0.8159	0.7040	0.7347	-0.0371	CNN
8	False	0	267	0.7950	0.7510	0.7655	-	CNN
9	True	1320	267	0.8201	0.7000	0.7323	-0.0332	CNN
10	False	0	229	0.7992	0.7154	0.7397	-	CNN
11	False	12000	229	0.7657	0.7121	0.7276	-0.0121	CNN
12	False	0	191	0.7657	0.7176	0.7319	-	CNN
13	False	12000	191	0.7113	0.7083	0.7092	-0.0227	CNN
14	False	12000	153	0.7322	0.7353	0.7343	0.0417	BiLSTM
15	False	0	153	0.6151	0.7313	0.6926	-	BiLSTM
16	False	12000	115	0.7490	0.6832	0.7001	0.0259	BiLSTM
17	False	0	115	0.7448	0.6544	0.6742	-	BiLSTM
18	False	0	77	0.6444	0.6968	0.6816	-	CNN
19	False	12000	77	0.6067	0.6872	0.6641	-0.0175	CNN
20	True	1320	39	0.7071	0.6190	0.6341	0.0056	BiLSTM
21	False	0	39	0.7782	0.6059	0.6285	-	BiLSTM
22	False	12000	0	0.1130	0.8438	0.4412	-	CNN

Table A.7: Results for Croatian source and English target. CNN on the left and BiLSTM on the right side. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1).

	CNN							BiLSTM					
	only_hate	#src	#tgt	r	p	F1		only_hate	#src	#tgt	r	p	F1
0	False	0	1523	0.8996	0.7264	**0.7772**	0	False	12000	1523	0.7615	0.7745	0.7699
1	True	1320	1523	0.8117	0.7549	0.7737	1	False	0	1523	0.7908	0.7560	0.7677
2	False	12000	1523	0.7573	0.7573	0.7573	2	True	1320	1523	0.7824	0.7305	0.7465
3	False	0	914	0.8285	0.7615	**0.7841**	3	False	12000	914	0.7950	0.7600	0.7718
4	False	12000	914	0.8326	0.7481	0.7754	4	False	0	914	0.8075	0.7539	0.7717
5	True	1320	914	0.7950	0.7570	0.7697	5	True	1320	914	0.7908	0.7441	0.7592
6	True	1320	610	0.8033	0.7471	**0.7654**	6	False	12000	610	0.7615	0.7615	0.7615
7	False	0	610	0.8368	0.7299	0.7623	7	True	1320	610	0.8326	0.7158	0.7494
8	False	12000	610	0.6862	0.7961	0.7541	8	False	0	610	0.7531	0.7469	0.7489
9	False	0	305	0.7448	0.7876	**0.7718**	9	False	12000	305	0.8075	0.7148	0.7416
10	True	1320	305	0.8159	0.7040	0.7347	10	False	0	305	0.8452	0.6733	0.7129
11	False	12000	305	0.7238	0.7300	0.7280	11	True	1320	305	0.7824	0.6751	0.7010
12	False	0	267	0.7950	0.7510	**0.7655**	12	False	12000	267	0.8159	0.7196	0.7478
13	True	1320	267	0.8201	0.7000	0.7323	13	True	1320	267	0.8117	0.6929	0.7239
14	False	12000	267	0.6360	0.7876	0.7297	14	False	0	267	0.8494	0.6465	0.6847
15	False	0	229	0.7992	0.7154	**0.7397**	15	False	12000	229	0.7197	0.7257	0.7238
16	False	12000	229	0.7657	0.7121	0.7276	16	False	0	229	0.8912	0.6283	0.6675
17	True	1320	229	0.7531	0.6950	0.7108	17	True	1320	229	0.8745	0.6239	0.6599
18	False	0	191	0.7657	0.7176	**0.7319**	18	False	12000	191	0.7657	0.6703	0.6930
19	False	12000	191	0.7113	0.7083	0.7092	19	False	0	191	0.8703	0.6246	0.6605
20	True	1320	191	0.7699	0.6815	0.7038	20	True	1320	191	0.8536	0.6145	0.6456
21	False	0	153	0.6778	0.7535	0.7273	21	False	12000	153	0.7322	0.7353	**0.7343**
22	False	12000	153	0.7531	0.6870	0.7043	22	False	0	153	0.6151	0.7313	0.6926
23	True	1320	153	0.7155	0.6602	0.6731	23	True	1320	153	0.7113	0.6296	0.6450
24	False	0	115	0.7699	0.6691	0.6928	24	False	12000	115	0.7490	0.6832	**0.7001**
25	False	12000	115	0.5816	0.7514	0.6906	25	False	0	115	0.7448	0.6544	0.6742
26	True	1320	115	0.7197	0.6772	0.6880	26	True	1320	115	0.7071	0.6426	0.6560
27	False	0	77	0.6444	0.6968	**0.6816**	27	False	12000	77	0.8285	0.6266	0.6585
28	False	12000	77	0.6067	0.6872	0.6641	28	True	1320	77	0.7280	0.6170	0.6351
29	True	1320	77	0.7238	0.6314	0.6487	29	False	0	77	0.5565	0.6584	0.6317
30	False	0	39	0.6987	0.6162	0.6301	30	True	1320	39	0.7071	0.6190	**0.6341**
31	True	1320	39	0.6695	0.6178	0.6270	31	False	0	39	0.7782	0.6059	0.6285
32	False	12000	39	0.8452	0.5906	0.6117	32	False	12000	39	0.8368	0.6024	0.6282
33	False	12000	0	0.1130	0.8438	**0.4412**	33	False	12000	0	0.0000	0.0000	0.3333

Table A.8: Compact result table for English source and Croatian target with ΔF1 scores. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1), and model used.

	only_hate	#src	#tgt	r	p	F1	ΔF1	model
0	False	0	1931	0.6494	0.7168	0.6981	-	CNN
1	False	10703	1931	0.8052	0.6596	0.6939	-0.0042	CNN
2	True	1196	1159	0.6851	0.6741	0.6795	0.0285	CNN
3	False	0	1159	0.5260	0.7074	0.6510	-	CNN
4	False	0	773	0.6948	0.6605	0.6713	-	BiLSTM
5	False	10703	773	0.6558	0.6667	0.6666	-0.0047	BiLSTM
6	True	1196	387	0.6591	0.6881	0.6826	0.0153	CNN
7	False	0	387	0.7857	0.6368	0.6673	-	CNN
8	False	0	338	0.6494	0.6826	0.6761	-	CNN
9	True	1196	338	0.5422	0.7357	0.6704	-0.0057	CNN
10	True	1196	290	0.6948	0.6903	0.6940	0.0161	CNN
11	False	0	290	0.6818	0.6731	0.6779	-	CNN
12	False	0	242	0.5812	0.7247	0.6794	-	CNN
13	True	1196	242	0.5877	0.7016	0.6691	-0.0103	CNN
14	True	1196	194	0.6429	0.6735	0.6680	0.0032	CNN
15	False	0	194	0.6364	0.6712	0.6648	-	CNN
16	True	1196	145	0.5877	0.6729	0.6522	0.0162	BiLSTM
17	False	0	145	0.6526	0.6281	0.6360	-	BiLSTM
18	False	10703	97	0.6104	0.5646	0.5727	0.2313	BiLSTM
19	False	0	97	0.0032	1.0000	0.3414	-	BiLSTM
20	False	0	49	0.6981	0.5556	0.5667	-	BiLSTM
21	False	10703	49	0.6623	0.5484	0.5577	-0.009	BiLSTM
22	False	10703	0	0.2403	0.6435	0.5083	-	CNN

Table A.9: Correlation matrix for metrics of English source and Croatian target.

	r	p	F1
#src	-0.159470	-0.238391	-0.078302
#tgt	0.292514	0.276780	0.487689

Table A.10: Results for English source and Croatian target. CNN on the left and BiLSTM on the right side. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1).

	CNN							BiLSTM					
	only_hate	#src	#tgt	r	p	F1		only_hate	#src	#tgt	r	p	F1
0	False	0	1931	0.6494	0.7168	**0.6981**	0	False	10703	1931	0.7110	0.6887	0.6972
1	False	10703	1931	0.8052	0.6596	0.6939	1	False	0	1931	0.7403	0.6766	0.6951
2	True	1196	1931	0.6656	0.6973	0.6906	2	True	1196	1931	0.6201	0.7127	0.6861
3	True	1196	1159	0.6851	0.6741	**0.6795**	3	True	1196	1159	0.6851	0.6635	0.6715
4	False	10703	1159	0.7565	0.6366	0.6624	4	False	0	1159	0.5714	0.7126	0.6697
5	False	0	1159	0.5260	0.7074	0.6510	5	False	10703	1159	0.6429	0.6689	0.6649
6	True	1196	773	0.5357	0.7143	0.6578	6	False	0	773	0.6948	0.6605	**0.6713**
7	False	0	773	0.6786	0.6451	0.6552	7	False	10703	773	0.6558	0.6667	0.6666
8	False	10703	773	0.6461	0.6525	0.6538	8	True	1196	773	0.6591	0.6424	0.6489
9	True	1196	387	0.6591	0.6881	**0.6826**	9	False	10703	387	0.6234	0.6621	0.6550
10	False	0	387	0.7857	0.6368	0.6673	10	False	0	387	0.5065	0.6996	0.6402
11	False	10703	387	0.6104	0.6573	0.6484	11	True	1196	387	0.5292	0.6626	0.6288
12	False	0	338	0.6494	0.6826	0.6761	12	False	0	338	0.6104	0.6738	0.6594
13	True	1196	338	0.5422	0.7357	**0.6704**	13	True	1196	338	0.5390	0.6721	0.6371
14	False	10703	338	0.6006	0.6584	0.6466	14	False	10703	338	0.5195	0.6723	0.6310
15	True	1196	290	0.6948	0.6903	**0.6940**	15	True	1196	290	0.5455	0.6857	0.6466
16	False	0	290	0.6818	0.6731	0.6779	16	False	0	290	0.6753	0.6303	0.6422
17	False	10703	290	0.5974	0.6502	0.6402	17	False	10703	290	0.6071	0.6192	0.6199
18	False	0	242	0.5812	0.7247	**0.6794**	18	False	0	242	0.5227	0.6910	0.6418
19	True	1196	242	0.5877	0.7016	0.6691	19	False	10703	242	0.6753	0.6265	0.6389
20	False	10703	242	0.6461	0.6258	0.6328	20	True	1196	242	0.5584	0.6165	0.6077
21	True	1196	194	0.6429	0.6735	**0.6680**	21	False	10703	194	0.5617	0.6528	0.6325
22	False	0	194	0.6364	0.6712	0.6648	22	True	1196	194	0.5519	0.6439	0.6243
23	False	10703	194	0.5357	0.6250	0.6081	23	False	0	194	0.5779	0.5993	0.5989
24	False	0	145	0.6818	0.6383	0.6503	24	True	1196	145	0.5877	0.6729	**0.6522**
25	True	1196	145	0.6396	0.6417	0.6441	25	False	0	145	0.6526	0.6281	0.6360
26	False	10703	145	0.3994	0.6758	0.5895	26	False	10703	145	0.5032	0.6225	0.5983
27	False	0	97	0.3604	0.6568	0.5668	27	False	10703	97	0.6104	0.5646	**0.5727**
28	True	1196	97	0.4058	0.5981	0.5582	28	False	0	97	0.0032	1.0000	0.3414
29	False	10703	97	0.3506	0.6102	0.5458	29	True	1196	97	0.9968	0.4968	0.3352
30	False	10703	49	0.4903	0.5763	0.5658	30	False	0	49	0.6981	0.5556	**0.5667**
31	False	0	49	0.7987	0.5395	0.5366	31	False	10703	49	0.6623	0.5484	0.5577
32	True	1196	49	0.4935	0.4780	0.4814	32	True	1196	49	1.0000	0.4960	0.3289
33	False	10703	0	0.2403	0.6435	**0.5083**	33	False	10703	0	0.0000	0.0000	0.3378

Table A.11: Compact result table for German source and Croatian target with ΔF1 scores. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1), and model used.

	only_hate	#src	#tgt	r	p	F1	ΔF1	model
0	False	1220	1931	0.7078	0.7219	0.7198	0.0145	BiLSTM
1	False	0	1931	0.7208	0.6959	0.7053	-	BiLSTM
2	False	1220	1159	0.6299	0.7321	0.7005	0.0495	CNN
3	False	0	1159	0.5260	0.7074	0.6510	-	CNN
4	False	1220	773	0.6039	0.7181	0.6838	0.0172	BiLSTM
5	False	0	773	0.6818	0.6583	0.6666	-	BiLSTM
6	True	610	387	0.6786	0.6677	0.6731	0.0095	CNN
7	False	0	387	0.7890	0.6328	0.6636	-	CNN
8	True	610	338	0.6591	0.6789	0.6762	0.0001	CNN
9	False	0	338	0.6494	0.6826	0.6761	-	CNN
10	False	0	290	0.6818	0.6731	0.6779	-	CNN
11	True	610	290	0.4643	0.7333	0.6380	-0.0399	CNN
12	False	0	242	0.5812	0.7247	0.6794	-	CNN
13	False	1220	242	0.6006	0.7034	0.6744	-0.005	CNN
14	False	0	194	0.6364	0.6712	0.6648	-	CNN
15	True	610	194	0.6558	0.6392	0.6457	-0.0191	CNN
16	False	0	145	0.6818	0.6383	0.6503	-	CNN
17	True	610	145	0.6526	0.6361	0.6425	-0.0078	CNN
18	True	610	97	0.6591	0.6006	0.6128	0.046	CNN
19	False	0	97	0.3604	0.6568	0.5668	-	CNN
20	False	0	49	0.6916	0.5635	0.5763	-	BiLSTM
21	True	610	49	0.5455	0.5773	0.5761	-0.0002	BiLSTM
22	False	1220	0	0.1851	0.5876	0.4684	-	BiLSTM

Table A.12: Correlation matrix for metrics of German source and Croatian target.

	r	p	F1
#src	-0.081869	0.093793	-0.079430
#tgt	0.199687	0.351221	0.517908

Table A.13: Results for German source and Croatian target. CNN on the left and BiLSTM on the right side. We report training type (only_hate is True when the source dataset contains only hate speech samples), number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1).

	CNN						BiLSTM					
	only_hate	#src	#tgt	r	p	F1	only_hate	#src	#tgt	r	p	F1
0	False	1220	1931	0.6591	0.7148	0.6999	False	1220	1931	0.7078	0.7219	**0.7198**
1	False	0	1931	0.6299	0.7239	0.6958	False	0	1931	0.7208	0.6959	0.7053
2	True	610	1931	0.8149	0.6005	0.6282	True	610	1931	0.7305	0.6579	0.6771
3	False	1220	1159	0.6299	0.7321	**0.7005**	False	1220	1159	0.6721	0.6571	0.6634
4	True	610	1159	0.5390	0.7186	0.6611	True	610	1159	0.5682	0.7028	0.6634
5	False	0	1159	0.5260	0.7074	0.6510	False	0	1159	0.5325	0.7193	0.6590
6	True	610	773	0.6948	0.6751	0.6827	False	1220	773	0.6039	0.7181	**0.6838**
7	False	0	773	0.6786	0.6451	0.6552	False	0	773	0.6818	0.6583	0.6666
8	False	1220	773	0.4740	0.7449	0.6466	True	610	773	0.6299	0.6599	0.6552
9	True	610	387	0.6786	0.6677	**0.6731**	True	610	387	0.7143	0.6395	0.6576
10	False	0	387	0.7890	0.6328	0.6636	False	0	387	0.5065	0.7059	0.6431
11	False	1220	387	0.7175	0.6406	0.6592	False	1220	387	0.5000	0.6814	0.6292
12	True	610	338	0.6591	0.6789	**0.6762**	False	0	338	0.6331	0.6610	0.6568
13	False	0	338	0.6494	0.6826	0.6761	False	1220	338	0.7532	0.6270	0.6522
14	False	1220	338	0.7013	0.6261	0.6430	True	610	338	0.5682	0.6554	0.6359
15	False	0	290	0.6818	0.6731	**0.6779**	False	0	290	0.7045	0.6420	0.6580
16	True	610	290	0.4643	0.7333	0.6380	False	1220	290	0.5552	0.6527	0.6306
17	False	1220	290	0.4448	0.7366	0.6309	True	610	290	0.6623	0.6126	0.6243
18	False	0	242	0.5812	0.7247	**0.6794**	True	610	242	0.5747	0.6969	0.6623
19	False	1220	242	0.6006	0.7034	0.6744	False	1220	242	0.4870	0.6667	0.6175
20	True	610	242	0.6721	0.6592	0.6651	False	0	242	0.8214	0.5776	0.5966
21	False	0	194	0.6364	0.6712	**0.6648**	False	0	194	0.5552	0.6381	0.6214
22	True	610	194	0.6558	0.6392	0.6457	False	1220	194	0.5455	0.6222	0.6086
23	False	1220	194	0.8571	0.5511	0.5499	True	610	194	0.9058	0.5397	0.5168
24	False	0	145	0.6818	0.6383	**0.6503**	True	610	145	0.4805	0.7081	0.6343
25	True	610	145	0.6526	0.6361	0.6425	False	1220	145	0.5487	0.6213	0.6088
26	False	1220	145	0.4675	0.6957	0.6240	False	0	145	0.7435	0.5783	0.5962
27	True	610	97	0.6591	0.6006	**0.6128**	True	610	97	0.3961	0.6703	0.5861
28	False	1220	97	0.5227	0.5730	0.5691	False	1220	97	0.4773	0.6000	0.5782
29	False	0	97	0.3604	0.6568	0.5668	False	0	97	0.0000	0.0000	0.3378
30	False	1220	49	0.4610	0.5992	**0.5737**	False	0	49	0.6916	0.5635	0.5763
31	False	0	49	0.7987	0.5395	0.5366	True	610	49	0.5455	0.5773	0.5761
32	True	610	49	0.5942	0.5027	0.5036	False	1220	49	0.6429	0.5485	0.5575
33	False	1220	0	0.9383	0.5026	0.4005	False	1220	0	0.1851	0.5876	**0.4684**

Table A.14: Full results for Multilingual BERT. CNN on the left and BiLSTM on the right side. We report number of source samples (#src), number of target samples (#tgt), recall (r), precision (p), F1 score (F1). In bold is the best result for every split.

	CNN						BiLSTM					
		#src	#tgt	r	p	F1		#src	#tgt	r	p	F1
en-de	0	0	766	0.7222	0.7109	**0.7165**	0	1523	766	0.7460	0.6667	**0.7041**
	1	1523	766	1.0000	0.4903	0.6580	1	0	766	0.7063	0.6593	0.6820
	2	1523	0	0.6587	0.5425	0.5950	2	1523	0	0.6508	0.6357	**0.6431**
en-hr	0	1523	1931	0.8117	0.6906	**0.7463**	0	0	1931	0.7435	0.7411	**0.7423**
	1	0	1931	0.0000	0.0000	0.0000	1	1523	1931	0.6526	0.7390	0.6931
	2	1523	0	0.1623	0.6250	0.2577	2	1523	0	0.4935	0.5779	**0.5324**
de-en	0	766	1523	0.8243	0.7269	0.7725	0	0	1523	0.8787	0.7167	**0.7895**
	1	0	1523	1.0000	0.5000	0.6667	1	766	1523	0.8703	0.7197	**0.7879**
	2	766	0	0.3766	0.6667	0.4813	2	766	0	0.6318	0.6741	**0.6523**
de-hr	0	766	1931	1.0000	0.4960	0.6631	0	766	1931	0.7597	0.7091	**0.7335**
	1	0	1931	0.0000	0.0000	0.0000	1	0	1931	0.7110	0.7526	**0.7312**
	2	766	0	0.5617	0.5986	0.5796	2	766	0	0.6721	0.6070	**0.6379**
hr-en	0	1931	1523	0.7573	0.7449	0.7510	0	0	1523	0.8494	0.7250	**0.7823**
	1	0	1523	1.0000	0.5000	0.6667	1	1931	1523	0.8368	0.7246	**0.7767**
	2	1931	0	0.0000	0.0000	0.0000	2	1931	0	0.7657	0.6399	**0.6971**
hr-de	0	0	766	0.7302	0.6866	**0.7077**	0	1931	766	0.7778	0.6203	**0.6901**
	1	1931	766	0.7857	0.5964	0.6781	1	0	766	0.6667	0.6512	0.6588
	2	1931	0	0.6587	0.5390	0.5929	2	1931	0	0.7143	0.5263	**0.6061**

Bibliography

[1] PyTorch-Transformers. URL `https://github.com/huggingface/pytorch-transformers`. [Online; accessed 9.9.2019].

[2] Croatian dataset by Styria Media Group AG, 2019.

[3] M. Artetxe and H. Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *CoRR*, abs/1812.10464, 2018.

[4] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma. Deep Learning for Hate Speech Detection in Tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760, 2017.

[5] C. Baziotis, N. Pelekis, and C. Doulkeridis. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[6] Bird, Steven and Klein, Ewan and Loper, Edward. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[7] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[8] U. Bretschneider and R. Peters. Detecting offensive statements towards foreigners in social media. In *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.

[9] F. Chollet et al. Keras. `https://keras.io`, 2015. [Online; accessed 29.8.2019].

[10] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[11] T. Davidson, D. Warmsley, M. Macy, and I. Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515, 2017.

[12] O. de Gibert, N. Pérez, A. G. Pablos, and M. Cuadros. Hate speech dataset from a white supremacy forum. *CoRR*, abs/1809.04444, 2018.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] B. Gambäck and U. K. Sikdar. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90, 2017.

[15] G. Glavas, R. Litschko, S. Ruder, and I. Vulic. How to (Properly) Evaluate Cross-Lingual Word Embeddings: On Strong Baselines, Comparative Analyses, and Some Misconceptions. *CoRR*, abs/1902.00508, 2019. URL `http://arxiv.org/abs/1902.00508`.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997. ISSN 0899-7667.

[17] Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[18] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL `http://www.scipy.org/`. [Online; accessed 29.8.2019].

[19] A. Joulin, P. Bojanowski, T. Mikolov, H. Jégou, and E. Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[20] Y. Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[24] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[25] S. Ruder, I. Vulić, and A. Søgaard. A survey of cross-lingual word embedding models. *arXiv preprint arXiv:1706.04902*, 2017.

[26] H. Saleem, K. Dillon, S. Benesch, and D. Ruths. A Web of Hate: Tackling Hateful Speech in Online Social Spaces. *arXiv preprint arXiv:1709.10159*, 2017.

[27] A. Schmidt and M. Wiegand. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, 2017.

[28] T. Schuster, O. Ram, R. Barzilay, and A. Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. 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 1599–1613. Association for Computational Linguistics, June 2019. URL `https://www.aclweb.org/anthology/N19-1162`.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf`.

[30] F. D. Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi, and M. Tesconi. Hate Me, Hate Me Not: Hate Speech Detection on Facebook. In *ITASEC*, 2017.

[31] W. Warner and J. Hirschberg. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26, 2012.

[32] Z. Waseem. Are you a racist or am I seeing things? Annotator influence on hate speech detection on Twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142, 2016.

[33] Z. Zhang and L. Luo. Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter. *Semantic Web*, Accepted (Preprint):1–21, 10 2018.