FRI Natural Language Processing course 2021
Project report

University *of Ljubljana*
Faculty *of Computer and Information Science*

Cross-Lingual Offensive Language Identification

Nikolina Grabovica, Selma Halilčević, Matjaž Mav

Abstract

In this paper we review a few publicly available datasets and a few different methods for offensive language identification. We explore traditional methods using handcrafted features, contextual embeddings and embedding alignment methods and current state of the art transformer models.

Keywords

Abusive content, offensive language, hate speech, social media, identification

Advisors: Slavko Žitnik

Introduction

Offensive language or hate speech is commonly understand as communication that expresses hate or encourages violence towards an individual or group based on some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic [1].

Existing offensive language prevention systems forced online writers to be more creative and these introduce additional challenges. Words and phrases may be obfuscated, for example 'a$$hole', 'ni99a' or 'kill yrslef'. Some of words or phrases might strongly relay on the context where they are used, for example word 'nigga' is associated with negative sentiment, but in some contexts this word may have neutral sentiment. Additionally, it is possible to compose hateful sentence with negations of neutral and positive words.

One of the challenges in research is to define offensive word. We define three broad types of offensive language – profanity, pejorative and obscenity. Profanity is a socially offensive language, which is also defined as cursing and swearing. It shows debasement of someone or something by using impolite, rude and culturally offensive language [2]. A pejorative or slur is the language expressing negative connotation, a low opinion or disrespect to someone or something. An obscenity is a dirty word or phrase expressing possible lewd, bawdy or offensive emotions to someone.

In this paper we review and explore the field of the offensive language detection, classification and clustering on different English datasets. The goal of this paper is to analyse a few of datasets and find more granular offensive language clusters among the top level classes that are commonly annotated in datasets.

The code and results are available on the Github repository matjazmav/fri-2021-nlp-project.

Related Work

In the most of early works simple features, like bag of words, n-grams and character n-grams are used. These simple features are reported to be highly predictive. Additionally Nobata C. et al. [3] show that other simple features like frequency of capitalization, use of non-numerical characters and links can also be used for the offensive language detection.

To detect offensive language in social media to prevent adolescent cyberbullying, Chen Y. et al. [4] propose the Lexical Syntactic Feature (LSF) architecture. This architecture assigns offensive weight value to each sentence by combining its offensiveness of words and the context. Words' offensiveness weight is calculated from its labeling type, which can be profanity, pejorative or obscenity. It is interesting to notice that combination of pejoratives and obscenities is labeled as strongly offensive if more than 80% of their use in a dataset is offensive, while being alone each is classified as a weak offensive word. To label a sentence context as offensive, LSF architecture captures all dependency types between words in the sentence and marks related words as intensifiers, which can describe users or other offensive words.

Another approach to prevent cyberbullying is proposed by Jacobs G. et al. [5] in which they derive positive and negative opinion word ratios and post polarity from subjectivity lexicon features. Subjectivity lexicons provide words' sentiments, which can express words' polarity (positive, negative or neutral), emotions or psychological processes.

Shen J. et al. [6] propose hierarchical clustering method known as Brown clustering. This clustering method has tendency to cluster words of opposite sentiment together, for

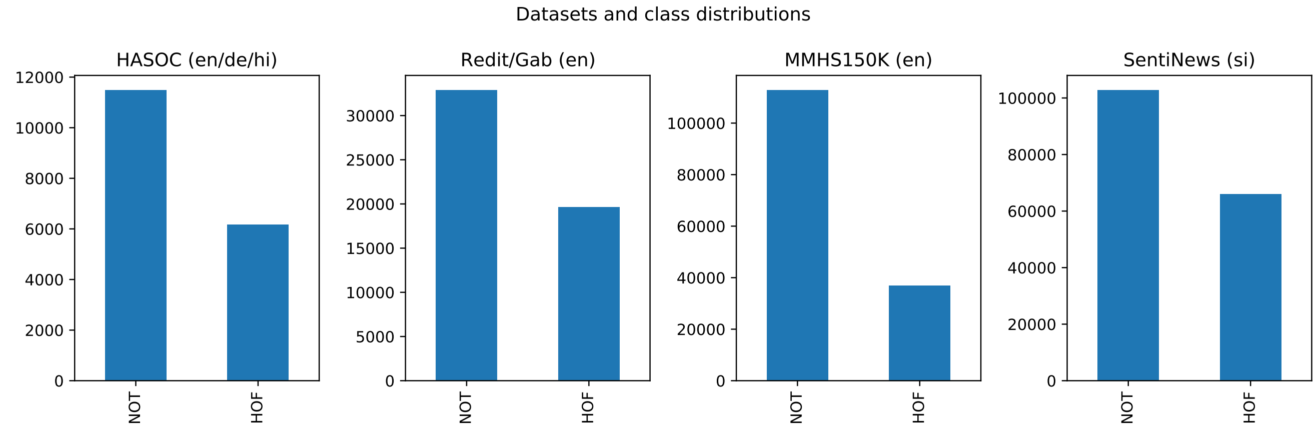


Figure 1. Comparison of class distributions between different datasets

example words like 'good' and 'bad' are clustered in the same cluster. In order to better generalize word representations Tian Z. et al. [7] apply Brown clustering method separately on the positive and negative sentiment data and later combine the information into a single complex feature. They show that the new information is beneficial to both simple and deep classifiers.

More recent research focus more on the deep learning methods. These deep representations of text (word, paragraph or document) are refereed to as embeddings. As we pointed out in the introduction, the context information of where the phrase is used is usually very important. The simplest approach to introduce the context information to the embedding is to average word embeddings of the entire phrase or sentence [3].

Martinc M. and Pollak S. [8] combine n-grams and convolutional neural network (CNN) for author profiling language variety classification. Inputs for the CNN are word bound character n-grams of sizes between three and five. They train six different classification models, where each model corresponds to one language group. Accuracy and $F1$-score for all language groups using TF-IDF are 0.9981 and 0.9981, respectively. They state that proposed system performed well for all binary predictions.

Datasets

To perform cross-lingual offensive language identification in social media, we test our methods on several online available datasets, listed below:

- HASOC [9] is a multilingual dataset composed of Twitter tweets and Facebook comments. It provides several thousands labeled social media posts for English, German and Hindi language.
- A Benchmark Dataset for Learning to Intervene in Online Hate Speech [10] contain a two fully-labeled datasets collected from Gab and Reddit in English language.
- MMHS150K [11] dataset containing English tweets that are annotated with multiple classes, such as racist, sexist, homophobic, religion based attack or attack to other community.
- SentiNews [12] dataset contains Slovene sentences that are annotated with the sentiment polarity. Content was scraped from different online sources and manually annotated.

The comparison of class distributions between different datasets is visualized in Figure 1.

HASOC

The HASOC dataset provides a few thousands labeled social media posts for English, German and Hindi language. The entire dataset was annotated and checked by the organizers of *the Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC)*. The provided annotations cover three distinct sub-tasks: (1) classification of hateful/offensive (HOF) and non-offensive (NOT) content, (2) more granular classification (HATE, OFFN and PRFN) and (3) targeted or un-targeted hateful/offensive language.

The HASOC dataset was, in addition, sampled from Twitter and partially from Facebook for all three languages. The Twitter API gives a large number of recent tweets which results in an objective dataset. Tweets were gathered using hashtags and keywords that contain offensive content. The collection was given to participants without metadata.

A Benchmark Dataset for Learning to Intervene in Online Hate Speech

The authors here provided two dataset one from Reddit and the another one from Gab both contain English conversations.

To extract high-quality conversational data that would probably include hate speech, they referenced the list of the ten most low-key toxic subreddits and collected data from subreddits: r/DankMemes, r/Imgoingtohellforthis, r/KotakuInAction, r/MensRights, r/MetaCanada, r/MGTOW, r/PussyPass, r/PussyPassDenied, r/The Donald, and r/TumblrInAc-

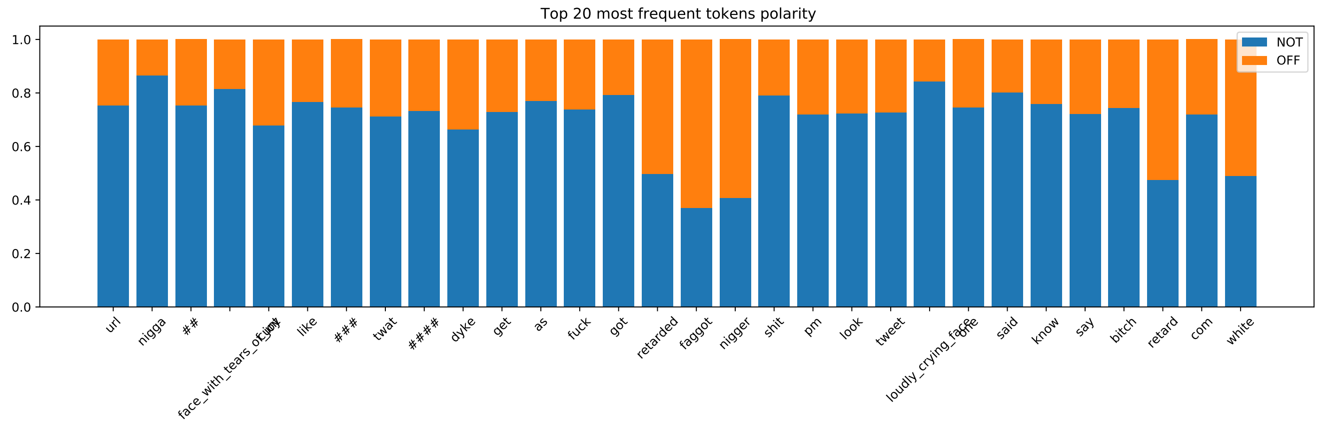


Figure 2. Top 20 most frequents tokens and its polarity

tion. They pulled out the top 200 newest submissions using Reddit's API. To further concentrate on conversations with hate speech in each submission, they use hate keywords to identify potentially hateful comments and then reconstruct the speaking context of each comment.

Data from all Gab posts is collected in October 2018. Similar to Reddit, they use hate keywords to pull out potentially hateful posts and rebuild the conversation context.

Multimodal Hate Speech Dataset

The Multimodal Hate Speech Dataset (MMHS150K) is manually annotated multimodal hate speech dataset composed by 150,000 tweets, each one of them contains text and an image. To gather real-time tweets from September 2018 until February 2019 the Twitter API was used, selecting the ones containing any of the 51 Hatebase (citation) terms that are common in tweets containing hate speech. Tweets that are containing less than three words were filtered out, and ones containing images are kept. To annotate gathered tweets, the crowd-sourcing platform Amazon Mechanical Turk was used. Workers were asked to classify the tweet text and image in one of six categories: no attacks to any community, racist, sexist, homophobic, religion based attacks or attacks to other communities.

SentiNews

The SentiNews dataset contains Slovene sentences annotated with the sentiment polarity values. Authors scraped the content from a few different online sources and manually annotated it.

We decide to use such dataset in the hateful/offensive language detection task since the negative polarity could be somewhat correlated to the hateful/offensive language. Additionally, other Slovene dataset are either not publicly available or some additional parsing is required (for example fetching tweets from the Twitter API).

We map negative sentiment values to hateful/offensive (HOF) class and neutral/positive to not-hateful/not-offensive (NOT) class.

Methods

In our experimental work we implemented 4 different approaches, all tackling the problem of hateful/offensive language detection. Some of the explored approaches are able to work in cross-lingual and multi-lingual setting.

Traditional models based on handcrafted features

First we implemented a simple model that is using handcrafted features. To extract features we first cleaned and tokenized sentences. Then we divided the training dataset into two chunks, samples annotated as hateful/offensive (OFF) and samples annotated as not hateful/offensive (NOT). From there we can compute token frequencies for each chunk (TF_{off} and TF_{not}) and for the whole training dataset (TF). From here we can define weighted term t frequencies as: $WTF_{off}(t) = TF_{off}(t)/TF(t)$ and $WTF_{not}(t) = TF_{not}(t)/TF(t)$. See Figure 2 where we visualized class polarity of top 20 tokens.

To obtain the final feature vector, list of tokens is further mapped into two vectors one is constructed from the corresponding $WTF_{off}(t)$ values and the other one from the corresponding $WTF_{not}(t)$ values. The final feature vector contains statistics (min, max, mean, median, std, ...) of this two vectors.

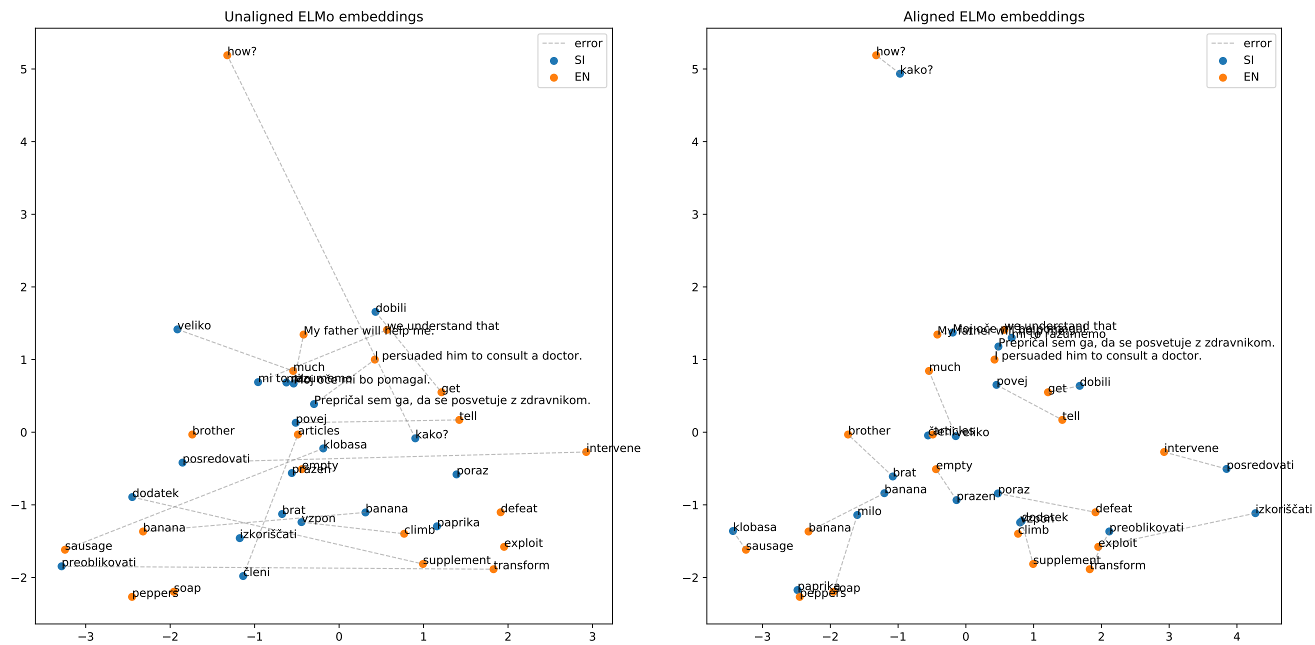
Finally, to make predictions Logistic Regression (LogReg) and Random Forest (RanFor) models are trained and evaluated along with the majority classifier (Dummy).

Traditional models based on ELMo embeddings

This approach is based on pre-trained ELMo [13] encoders for English and Slovene language [14, 15].

The final feature vector is just an average of all ELMo embedding vectors for the corresponding tokens in a sentence. To make predictions, Logistic Regression (LogReg) and Random Forest (RanFor) models are trained and evaluated.

To make this approach cross-lingual we need to find a mapping from Slovene ELMo embedding vector space to English ELMo embedding vector space. Here we used simple Linear Regression model to learn the mapping function. The



(a) Unaligned ELMo embedding pairs (SI-EN) **(b)** Aligned ELMo embedding pairs (SI-EN)

Figure 3. Comparison between unaligned and aligned ELMo embedding pairs (EN-SI). PCA was used to obtain the visualizations.

comparison between unaligned and aligned embeddings is visualized on the Figure 3.

The data that we use to learn the mapping function was manually obtained from the http://mylanguages.org/learn_slovenian.php and http://www.manythings.org/anki/.

Transfer learning of BERT models

Here we used pre-trained BERT [16] (bert-base-uncased) and multilingual mBERT (bert-base-multilingual-uncased) models from the HuggingFace [17] repository and then fine tuned them for the binary classification task. To balanced out the class imbalance we weighted loss values with the inverse class distribution. We train each model on English dataset and on multilingual dataset (English, German, Hindu and Slovenian).

Transfer learning of T5 models

Here we use pre-trained T5 [18] (t5-small) and multilingual mT5 [19] (google/mt5-small) models from the HuggingFace [17] repository and then fine tune them for the binary classification task. We use random under sampling method to balance out the class imbalances. We train each model on English dataset and on multilingual dataset (English and Slovenian).

Results & Discussion

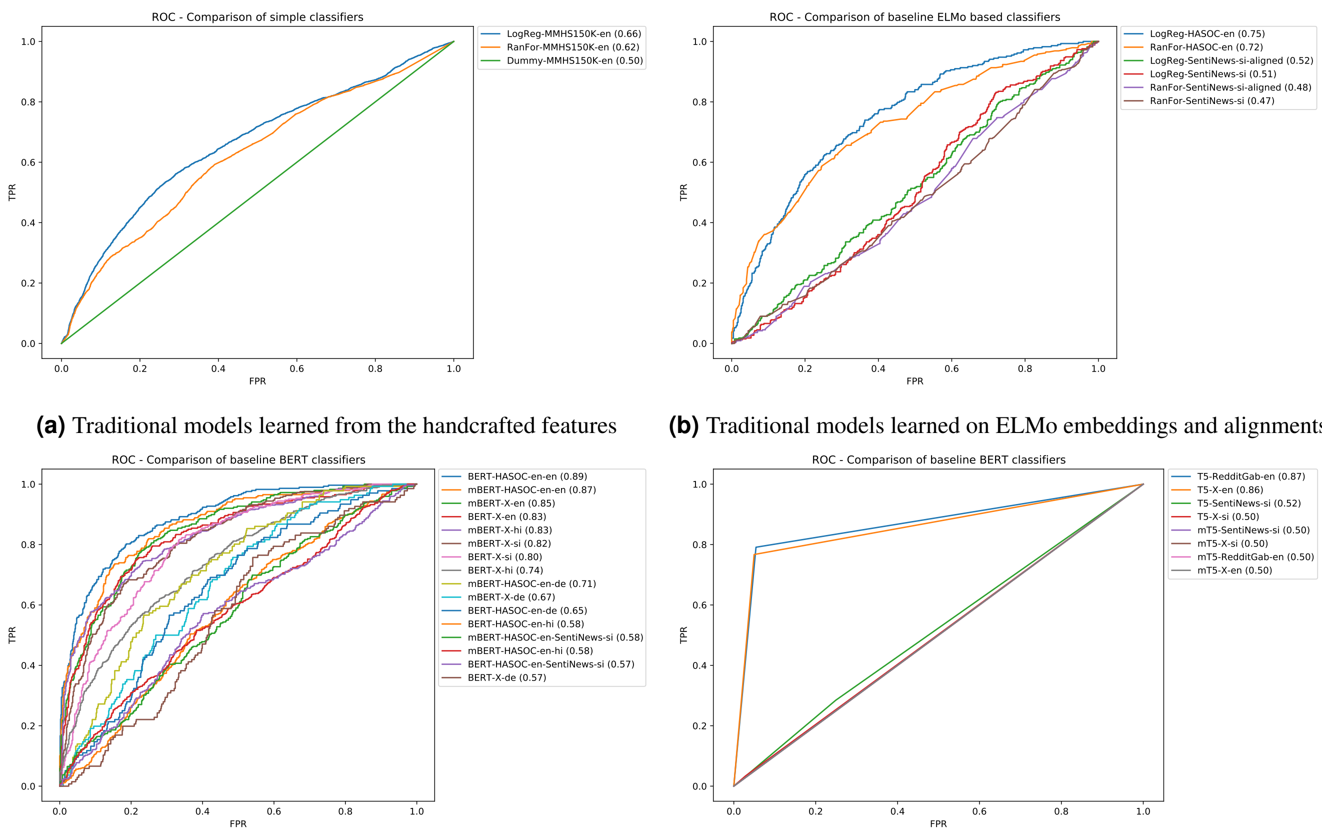
We evaluate previously described models on different combination of training and evaluation dataset. Table 1 lists evaluation metrics of all our experiments where we also highlight the best model for each language. In the Figure 4 we visualize ROC curves for the experiments.

As expected, transformer models achieve better scores then the more traditional methods. However, finding the right set of hyper parameters was tricky. It is interesting that the observed training loss of the BERT and the mBERT models do not converge but the T5 and the mT5 models do. Unfortunately, we are unable to get meaningful predictions from the mT5 model.

Conclusion

In our work we explore different approaches for hateful/offensive language detection. We explore more simple & traditional approaches and also some of the current state of the art approaches. We describe and use a few datasets and a few different languages to train and evaluate the implemented models.

It would be interesting if we could get mT5 model working. Also, aligning embedding vectors seems relatively simple and could provide a fairly good results, but more advanced alignment method should be used to conclude that.



(a) Traditional models learned from the handcrafted features

(b) Traditional models learned on ELMo embeddings and alignments

(c) Transfer learning of BERT and mBERT models

(d) Transfer learning of T5 and mT5 models

Figure 4. Evaluation results visualized as ROC curves of 4 different approaches. Legends are sorted by AUC values. Model naming: *<model_type>— <learning_dataset>—<prediction_language>*

Model	F1	Pr	Re	AUC	Model	F1	Pr	Re	AUC
LogReg-MMHS150K-en	.46	.38	.57	.66	BERT-HASOC-en-en	.64	.51	.86	.89
RanFor-MMHS150K-en	.32	.44	.25	.62	BERT-HASOC-en-de	.24	.22	.27	.65
Dummy-MMHS150K-en	.00	.00	.00	.50	BERT-HASOC-en-hi	.01	.50	.01	.58
LogReg-HASOC-en	.51	.37	.79	.75	BERT-HASOC-en-SentiNews-si	.08	.45	.04	.57
LogReg-SentiNews-si	.07	.29	.04	.51	mBERT-HASOC-en-en	.62	.50	.83	.87
LogReg-SentiNews-si-aligned	.39	.41	.38	.52	**mBERT-HASOC-en-de**	**.39**	.29	.59	.71
RanFor-HASOC-en	.44	.57	.36	.72	mBERT-HASOC-en-hi	.38	.57	.28	.58
RanFor-SentiNews-si	.22	.34	.16	.47	mBERT-HASOC-en-SentiNews-si	.05	.67	.02	.58
RanFor-SentiNews-si-aligned	.02	.27	.01	.48	BERT-X-en	.63	.54	.75	.83
T5-RedditGab-en	.84	.90	.79	.87	BERT-X-de	.00	.00	.00	.57
T5-SentiNews-si	.35	.45	.28	.52	BERT-X-hi	.68	.56	.84	.74
T5-X-en	**.83**	.90	.77	.86	BERT-X-si	.67	.61	.75	.80
T5-X-si	.06	.46	.03	.50	mBERT-X-en	.57	.42	.89	.85
mT5-RedditGab-en	.00	.00	.00	.50	mBERT-X-de	.04	.21	.02	.67
mT5-SentiNews-si	.00	.00	.00	.50	**mBERT-X-hi**	**.73**	.63	.86	.83
mT5-X-en	.00	.00	.00	.50	**mBERT-X-si**	**.68**	.59	.80	.82
mT5-X-si	.00	.00	.00	.50					

Table 1. Evaluation results of 4 different approaches. Bolded models shown best results (using the F1 measure) for the corresponding language.
Model naming: *<model_type>— <learning_dataset>—<prediction_language>*

References

[1] Anna Schmidt and Michael 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.

[2] *The Monthly Supplement: a current biographical reference service*. Number v. 1-2. A. N. Marquis., 1940.

[3] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153, 2016.

[4] Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 71–80. IEEE, 2012.

[5] Gilles Jacobs, Cynthia Van Hee, and Véronique Hoste. Automatic classification of participant roles in cyberbullying: Can we detect victims, bullies, and bystanders in social media text? *Natural Language Engineering*, pages 1–26, 2020.

[6] Judong Shen, Shing I Chang, E Stanley Lee, Youping Deng, and Susan J Brown. Determination of cluster number in clustering microarray data. *Applied Mathematics and Computation*, 169(2):1172–1185, 2005.

[7] Zuoyu Tian and Sandra Kübler. Offensive language detection using brown clustering. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5079–5087, 2020.

[8] Matej Martinc and Senja Pollak. Combining n-grams and deep convolutional features for language variety classification. *Natural Language Engineering*, 25(5):607–632, 2019.

[9] Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. FIRE '19, page 14–17, New York, NY, USA, 2019. Association for Computing Machinery.

[10] Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. A benchmark dataset for learning to intervene in online hate speech. *arXiv preprint arXiv:1909.04251*, 2019.

[11] Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas. Exploring hate speech detection in multimodal publications. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1470–1478, 2020.

[12] Jože Bučar, Martin Žnidaršič, and Janez Povh. Annotated news corpora and a lexicon for sentiment analysis in slovene. *Language Resources and Evaluation*, 52(3):895–919, 2018.

[13] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[14] Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[15] Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 271–276, Gothenburg, Sweden, May 2017. Association for Computational Linguistics.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional trans-

formers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[17] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[18] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[19] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.