STUFIIT at SemEval-2019 Task 5: Multilingual Hate Speech Detection on Twitter with MUSE and ELMo Embeddings

Michal Bojkovský
Faculty of Informatics and Information Technologies STU in Bratislava
Ilkovičova 2, Bratislava
m.bojkovsky@protonmail.com

Matúš Pikuliak
Faculty of Informatics and Information Technologies STU in Bratislava
Ilkovičova 2, Bratislava
matus.pikuliak@stuba.sk

Abstract

We evaluate the viability of multilingual learning for the task of hate speech detection. We also experiment with adversarial learning as a means of creating a multilingual model. Ultimately our multilingual models have had worse results than their monolignual counterparts. We find that the choice of word representations (word embeddings) is very crucial for deep learning as a simple switch between MUSE and ELMo embeddings has shown a 3-4% increase in accuracy. This also shows the importance of context when dealing with online content.

1 Introduction

The Internet has been surging in popularity as well as general availability. This has considerably increased the amount of user generated content present online. This has, however, brought up a few issues. One of the issues is hate speech detection, as manual detection has been made nearly impossible by the quantity of data. The only real solution is automated hate speech detection. Our task is detection of hate speech towards immigrants and women on Twitter (Task A).

Hate speech can be defined as "Any communication that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics." (Basile et al., 2019) This proves to be a very broad definition, because utterances can be offensive, yet not hateful (Davidson et al., 2017). Even manual labeling of hate speech related data is notoriously difficult as hate speech is very subjective in nature (Nobata et al., 2016; Waseem, 2016).

The provided dataset consists of collected messages from Twitter in English or Spanish language. Hate speech datasets are very prone to class imbalances (Schmidt and Wiegand, 2017). The provided dataset does not suffer from this problem. The English data contains 10,000 messages with 42.1% of the messages labeled as hate speech. The Spanish data contains 4969 messages and similarly to the English part, 41.5% were labeled as hate speech. This gives us a dataset with 14969 messages of which 6270 are categorized as hate-speech. We have not used any additional sources of training data for our models. More information about the data can be found in the Task definition (Basile et al., 2019).

Most research dealing with hate speech has been done in English due to labelled dataset availability. However, this issue is not unique to English-based content. In our work, we explore multilingual approaches, as we recognize data imbalance between languages as one of major challenges of NLP. Multilingual approaches could help remedy this problem, as one could transfer knowledge from a data-rich language (English) to a data-poor language (Spanish).

1.1 Background

We focus on neural network approaches, as they have been achieving better performance than traditional machine learning algorithms (Zhang et al., 2018). We explore both monolingual and multilingual learning paradigms. Multilingual approaches enable us to use both English and Spanish datasets for training.

The most popular input features in deep learning are word embeddings. Embeddings are fixed length vectors with real numbers as components, used to represent words in a numeric way. The input layers to our models consist of MUSE (Conneau et al., 2017) or ELMo (Peters et al., 2018) word embeddings.

MUSE embeddings are multilingual embeddings based on fastText. They are available in different languages, where the words are mapped into

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 464–468
Minneapolis, Minnesota, USA, June 6–7, 2019.

the same vector space across languages, i.e. words with similar meanings across languages have a similar vector representation.

ELMo provide a deep representation of words based on output of a three layer pre-trained neural network. The representation for a word is based on the context in which the word is used. However, they are not multilingual representations.

To work around the monolinguality of ELMo, we use a technique called *adversarial learning* (Ganin and Lempitsky, 2014). Adversarial networks consist of three parts:

- *Feature extractor* responsible for creating representations belonging to the same distribution regardless of input data distribution i.e. of the language the messages are in. This transformation is learned during training.
- *Classifier* responsible for the classification i.e. labeling hateful utterances.
- *Discriminator* responsible for predicting the language of a given message.

During backpropagation, the loss from classifier (L_{cls}) is computed the standard way. The loss from discriminator (L_{dis}) has its sign flipped and is multiplied by *adversarial lambda* (λ). The discriminator works *adversarialy* to the classificator.

$$Loss = L_{cls} - \lambda L_{dis} \quad (1)$$

The loss from the discriminator encourages the feature extractor to create indistinguishable representations for messages across languages. This is most often implemented by a gradient reversal layer.

2 Implementation details

2.1 Preprocessing

Traditionally, neural network models have a very simple preprocessing pipeline. However, internet communication is very bloated (URLs, mentions, emoji etc.). As such we have decided to remove all the noise from the messages.

At first, we remove URLs and name mentions from messages. These contain no useful information for our prediction. Afterwards, we transform malformed markup characters such as *&gt* into their one character representations (>). We also remove the hash symbol from hashtags as it can be problematic for tokenizers to work with. Next we employ demojization. We use a Python library called *Emoji*[1]. For example, this let us change the unicode representation of a thumbs up emoji into :thumbs_up:, which is then parsed into usable text 'thumbs up'. The next step is tokenization and stop words removal. For this step, we use a library called *spaCy*[2]. We chose this library as it has support for both English and Spanish and we aim to have the same preprocessing pipeline for different languages. We also remove lone standing non-alphanumeric characters, which are often found after tokenization. As the last few steps, we change all characters into lowercase, change numbers into a number token. Sentence size is limited to 64. This was enough for nearly all of the tweets after preprocessing.

2.2 Tested architectures

For MUSE, we use pretrained embeddings made available by Facebook research. We also use pretrained ELMo representations (Che et al., 2018; Fares et al., 2017), which support English as well as Spanish. Both can be found on GitHub [3] [4]. The embeddings were not modified during training.

We examine two different model architectures: LSTM based one and a CNN+LSTM hybrid. The combination of two learning paradigms, two model architectures and two different input representations sum up to 8 different models. All of the models use cross-entropy as the loss function.

2.2.1 Monolingual approaches

Monolingual models were used and trained independently on English and Spanish parts of the dataset.

LSTM-based approach

We use both word-level and char-level representations with ELMo. The representations are then independently fed into a bidirectional LSTM layer of size 64. The output of each of these layers is then fed into an attention layer.

Next, the outputs are concatenated into a single vector and used as an input of a fully connected layer with 20 cells with ReLU activation function. The last layer is a softmax layer with L1 and L2 regularization used for final predictions. The output is then the probability of classes for predicted

[1] https://pypi.org/project/emoji/
[2] https://spacy.io/
[3] https://github.com/facebookresearch/MUSE
[4] https://github.com/HIT-SCIR/ELMoForManyLangs

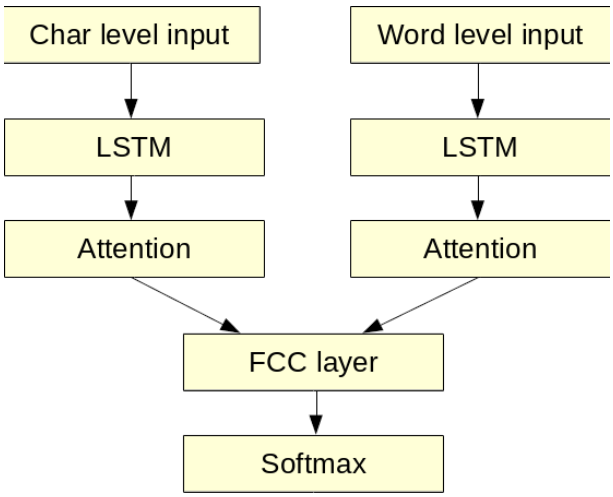

Figure 1: LSTM-based ELMo monolingual model

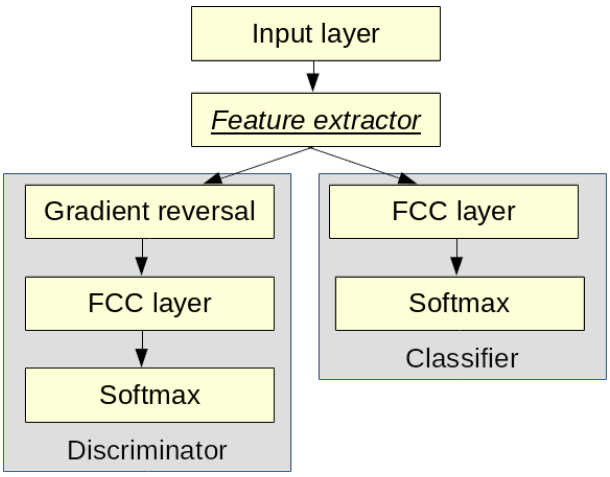

Figure 2: Base adversarial model

variable (non hate speech or hate speech). The model can be seen on Figure 1.

For MUSE, we have only word-level information available. As we have only one input, we only need one LSTM and attention layer. Otherwise, the models are the same.

CNN-based approach

The input layer is fed into a convolutional layer. This layer performs a 1d convolution with 100 filters and a kernel size of 4 with a relu activation function. This is then max pooled with a pool size of 4 and stride of 4. These layers can be understood as a feature extrator part of the model. These extracted features are then fed into a monodirectional LSTM layer with size of 64. The output is global max pooled and fed into the last softmax layer. For ELMo we have used the average representation of all its layers.

2.2.2 Multilingual approaches

Multilingual models were trained on concatenated English and Spanish data.

Multilingual MUSE models

With MUSE embeddings a multilingual approach is straightforward. We use both the approaches previously mentioned (LSTM and CNN+LSTM) without any further changes, as they are implicitly multilingual.

Multilingual ELMo models

The base architecture of our model can be seen on Figure 2. After the input layer is a feature extractor. We have used either an LSTM with attention or a 1d convolutional layer with max-pooling as described in previous sections. The discriminator and classifier include a single FCC layer with a final softmax layer in both cases. The FCC layers have 32 cells each.

The difference between them is the presence of a gradient reversal layer in the discriminator. The gradient is multiplied by -0.25 during backpropagation. This value for adversarial lambda was found empirically. Both the classifier and discriminator were trained simultaneously.

Architecture	Acc	Rec	Prec	F1
$\text{LSTM}_{\text{mono-elmo}}$	**0.733**	0.676	0.697	**0.683**
$\text{CNN}_{\text{mono-elmo}}$	0.69	0.698	0.640	0.655
$\text{LSTM}_{\text{mono-muse}}$	0.675	0.694	0.606	0.645
$\text{CNN}_{\text{mono-muse}}$	0.658	**0.761**	0.577	0.655
$\text{LSTM}_{\text{multi-elmo}}$	0.695	0.386	**0.799**	0.517
$\text{CNN}_{\text{multi-elmo}}$	0.673	0.448	0.693	0.52
$\text{LSTM}_{\text{multi-muse}}$	0.664	0.677	0.594	0.632
$\text{CNN}_{\text{multi-muse}}$	0.661	0.677	0.59	0.632

Table 1: Results on English dataset (Task A)

3 Results evaluation

We show detailed results in both English (Table 1) and Spanish (Table 2). We use a subscript of *mono* or *multi* to differentiate between learning methods and *muse* or *elmo* to differentiate between architectures in the table. The table was completed by computing the mean of 5 runs of each model on the validation part of the datasets. The validation set consisted of 10% available data. Multilingual models were trained with concatenated English and Spanish datasets.

None of the multilingual models were able to

Architecture	Acc	Rec	Prec	F1
$LSTM_{mono\text{-}elmo}$	**0.768**	**0.742**	**0.748**	**0.738**
$CNN_{mono\text{-}elmo}$	0.726	0.65	0.726	0.657
$LSTM_{mono\text{-}muse}$	0.711	0.712	0.662	0.689
$CNN_{mono\text{-}muse}$	0.72	0.731	0.673	0.699
$LSTM_{multi\text{-}elmo}$	0.556	0.123	0.419	0.173
$CNN_{multi\text{-}elmo}$	0.588	0.332	0.712	0.345
$LSTM_{multi\text{-}muse}$	0.723	0.701	0.684	0.692
$CNN_{multi\text{-}muse}$	0.718	0.688	0.681	0.685

Table 2: Results on Spanish dataset (Task A)

outperform the baseline monolingual LSTM based model with ELMo. Not even in a multilingual setting of averaging results between languages. Multilingual MUSE has not shown any significant increase in performance compared to monolingually trained MUSE.

The results show how potent ELMo embeddings are. Online content can often be offensive and vulgar, while still being non-hateful. This is often enough for a model to classify an utterance as hate speech (Davidson et al., 2017; Hemker, 2018). In these situations, ELMo has an advantage, as the representations are built entirely in the context of a sentence as a whole.

The adversarial models achieved the worst performance. On first glance, judging by accuracy, the models seem to perform on a very average level. After further analysis, we can see that their performance was very poor and inconsistent, e.g.the LSTM based model achieved only 0.123 recall on spanish dataset. The model labeled only a few messages as hate speech and even those not very successfully. The relatively high accuracy was a result of data distribution, as 55.6% of the data was non-hate speech.

We can also see that only in this category the CNN based models outperformed LSTM based models. This implies that for adversarial learning to work, one has to use a very robust feature extractor. It is also the only time that the performance on English was higher than on Spanish. This is the result of data scarcity, as the extractor had a hard time creating truly multilingual representations. This could also be seen during training as the discriminator hovered around 90% accuracy.

For our task submission, we have used the monolingual LSTM model based on ELMo, which we considered as our baseline model. We have achieved results shown in Table 3.

Language	Acc	Pre	Rec	F1	Place
English	0.47	0.59	0.54	0.42	44
Spanish	0.71	0.7	0.7	0.7	18

Table 3: Results on test dataset

4 Conclusion and future work

In this paper we have evaluated a few simple neural network models in a monolingual and multilingual context. We have included our unsuccessful models to inspire further research in this direction.

We conclude that the quality of word representations used has a significant impact on the performance of a model. Changing between MUSE and ELMo resulted in a 3 - 4% increase in accuracy even when MUSE based models could benefit from multilingual training. The contextual nature of ELMo representations make them much more flexible and less domain constrained than traditional word embeddings. Simple models (as the one we proposed) are able to achieve decent results this way. We can also see that using adversarial learning needs a lot of available data to be at all viable.

We believe that more research should be put into multilingual solutions. The feature extractor needs more training data to create truly ambiguous representations of utterances between languages. We will look into testing our model with more training data to evaluate the value of adversarial learning for multilingual hate speech detection or pretraining the feature extractor on a different task with more data available.

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Rangel, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*. Association for Computational Linguistics", location = "Minneapolis, Minnesota.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. 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. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. 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. Association for Computational Linguistics.

Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised Domain Adaptation by Backpropagation. (i).

Konstantin Hemker. 2018. Data Augmentation and Deep Learning for Hate Speech Detection.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive Language Detection in Online User Content. *Proceedings of the 25th International Conference on World Wide Web - WWW '16*, pages 145–153.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection using Natural Language Processing. *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, (2012):1–10.

Zeerak Waseem. 2016. 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. Association for Computational Linguistics.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting Hate Speech Using a Convolution-GRU Based Deep Neural Network. 7185(June).