CHILab @ HaSpeeDe 2: Enhancing Hate Speech Detection with Part-of-Speech Tagging

Giuseppe Gambino and Roberto Pirrone

DOI: 10.4000/books.aaccademia.7057
Publisher: Accademia University Press
Place of publication: Torino
Year of publication: 2020
Published on OpenEdition Books: 11 May 2021
Serie: Collana dell'Associazione Italiana di Linguistica Computazionale
Electronic ISBN: 9791280136329

http://books.openedition.org

Electronic reference
CHILab @ HaSpeeDe 2: Enhancing Hate Speech Detection with Part-of-Speech Tagging

Giuseppe Gambino and Roberto Pirrone
Dipartimento di Ingegneria
Università degli Studi di Palermo
giuseppe.gambino09@community.unipa.it
roberto.pirrone@unipa.it

Abstract
The present paper describes two neural network systems used for Hate Speech Detection tasks that make use not only of the pre-processed text but also of its Part-of-Speech (PoS) tag. The first system uses a Transformer Encoder block, a relatively novel neural network architecture that arises as a substitute for recurrent neural networks. The second system uses a Depth-wise Separable Convolutional Neural Network, a new type of CNN that has become known in the field of image processing thanks to its computational efficiency. These systems have been used for the participation to the HaSpeeDe 2 task of the EVALITA 2020 workshop with CHILab as the team name, where our best system, the one that uses Transformer, ranked first in two out of four tasks and ranked third in the other two tasks. The systems have also been tested on English, Spanish and German languages.

1 Introduction
Hate speech is not unfortunately a new problem in the society, but recently it has found fertile ground in social media platforms that enable users to express themselves freely and often anonymously. While the ability to freely express oneself is a human right, inducing and spreading hate towards another group is an abuse of this liberty (MacAvaney et al., 2019).

As such, many online micro-blogs such as Facebook, YouTube, Reddit, and Twitter consider hate speech harmful, and have both policies and instruments to remove hate speech content, that are getting better over time. Due to the societal concern and how widespread hate speech is becoming on the Internet, there is strong motivation to study automatic detection of hate speech. By doing so, the spread of hateful content can be reduced, having a safer place to stay online for the community but also a more attractive place for advertising sponsors who do not want their brand to be associated with hateful content. Obviously, detecting hate speech is a challenging task. For example, in case of wrong classification, a content creator could suffer socio-economic consequences such as the demonetization of one of its contents or the ban from the platform used. Therefore, the goal of hate speech detection is not only to identify a text that contains words that at first sight could be negative, but also to be able to distinguish news headlines that talk about crime news from a text that contains an effective “attack” against a person or group on the basis of attributes such as race, religion, ethnic origin, national origin, sex, disability, sexual orientation, or gender identity.

The rest of the paper is arranged as follows. Section 2 reports a description of our systems developed for hate speech detection tasks. Section 3 shows the results obtained in the HaSpeeDe 2 (Sanguinetti et al., 2020) task of the EVALITA 2020 (Basile et al., 2020) conference, together with other results obtained with different languages. Results are showed in Section 4 and conclusions are discussed in Section 5.

2 Description of the Systems
In this section we present the implementation details of all the used architectures. Both the systems we implemented share the use of PoS Tagging technique that is applied to the pre-processed text, and passed as an additional input to the neural network.
2.1 Pre-processing
Before training a model, it is common practice to clean the data, especially if they are retrieved from social media. For this reason we implemented a classic text pre-processing pipeline, that consists of: lower casing the text; removing HTML tags, mention and symbols; standardizing words by cutting the characters repeated more than two times in a row. We also made some keyword substitutions in all our data sets:

- URLs and the “url” keyword of the HaSpeeDe 2 data set were replaced by the symbol LINKURL
- Happy emoticons like “:)” or “:D” were replaced by the symbol HAPPYEMO
- Angry or sad emoticons like “:@” or “:(” were replaced by the symbol BADEMO

It is important to note that we have not removed the emojis from the text as our word embedding takes into account emojis as plain words.

2.2 Part-of-Speech Tagging
In this work we use the PoS Tagging technique to provide our networks with more information about the meaning of a sentence through an explicit classification on the basis of its grammatical structure. This is a crucial point with regards to hate sentences. In fact they tend to have particular structures. As an example, one of the most widespread hate sentence is the verbless one, also known as nominal utterance (Comandini et al., 2018). Another example are journalistic tweets (Comandini and Patti, 2019). Starting from a preliminary direct inspection of the development data set proposed in HaSpeeDe 2, we found that usually a journalistic tweet is a short tweet that ends with an URL. Such texts can be easily misclassified due to the presence of some negative words that explain the news. Table 1 reports some examples of these types of statements.

As the HaSpeeDe 2 organizers required explicitly to use the same system for both tasks A and B, we set up a PoS Tagging model not too biased towards either news headlines or tweets. As a consequence, we enriched the PoS Tagger provided by the Python’s spaCy library (Honnibal and Montani, 2017). As this model is trained on Wikipedia, we used some regex formulas to add the keywords for emoticons, emojis, hashtags, and URLs to the vocabulary. In this way we have injected some parts of the speech of the social media language into a standard PoS Tagging model. We were definitely aware that tweet oriented models such as UDpipe tool (Straka, 2018) trained on POSTWITA-UD Treebank (Sanguinetti et al., 2018) would have performed better than our solution on the in-domain data but our solution guaranteed a more balanced performance. An example of our PoS Tagging is showed in Figure 1.

2.3 Word Embedding
It is well known in the NLP community that word embeddings are one of the features that most affects the performance of a model.

For our application we chose fastText (Bojanowski et al., 2016), a word embedding developed by Facebook Research. FastText enriches word vectors with subword information treating each word as composed of n-grams. Each word vector is the sum of the vector representations of each of its n-grams. In this way, two words not only will have nearby vectors if they have similar context but also if they are similar. This is a great feature to treat miss-spelling that occurs of-

<table>
<thead>
<tr>
<th>Tweet</th>
<th>HS</th>
</tr>
</thead>
<tbody>
<tr>
<td>@user useless people like all Muslims</td>
<td>1</td>
</tr>
<tr>
<td>@user no more refugees in Italy please no more</td>
<td>1</td>
</tr>
<tr>
<td>Four bicycles stolen from Milan-Sanremo cyclists: found in a gypsy camp url</td>
<td>0</td>
</tr>
<tr>
<td>TRAGEDY IN PRISON - The nomad Carlo Helt takes his own life url</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: Some examples translated into English drawn from the development data set proposed in the HaSpeeDe 2 competition together with their label: nominal utterances used in hate speech along with journalistic tweets
ten in social languages. We trained from scratch the word embedding for the Italian language with the Gensim library (Rehurek and Sojka, 2010) on a 2014 MacBook Pro 13” with 8GB RAM and AVX2 FMA CPU extension and it took about 5 hours. The embedding model has been trained for 10 epochs on 5 millions Italian tweets, with a size = 300, window_size = 5, and min_count = 2. These tweets were extracted from TWITTA 2018 Dataset (Basile and Nissim, 2013) and are all related to the words: immigrati, islam, migranti, musulmani, profughi, rom, stranieri, salvini, criminali, africani, terroni, #dallavostraparte, #salvini, #stopinvasione, #piazzapulita, #quintacolonna.

For the French, English and German tweets we used pre-trained models (Camacho-Collados et al., 2020). Regarding the PoS Tagging embedding, we have applied the TensorFlow’s Embedding Layer for all the languages considered.

2.4 System 1: The Transformer

Transformers (Vaswani et al., 2017) are the current state-of-the-art models for dealing with sequences. Unlike previous architectures for NLP, such as LSTM and GRU, there are no recurrent connections and thus no real memory of previous states. Transformers get around this lack of memory by perceiving entire sequences simultaneously and treating them with an attention mechanism. In this way, Transformers achieve parallelism that leads to a significantly shorter training time than recurrent solutions. Attention is a means of selectively weighting different elements in input data, so that they will have an adjusted impact on the hidden states of downstream layers.

A Transformer was conceived as an encoder-decoder model, that is an ideal approach for machine translation tasks and language modeling. In this work we used the Transformer encoder architecture, as an alternative to recurrent or convolutional neural networks (CNN) (see Figure 2). We used just one Transformer encoder for the text input and one for the PoS input, then we averaged them thorough max pooling. Finally, we used dropout and a dense layer to get the output probabilities. After testing various combinations of parameters, we found that the most efficient for this task are: 12 heads in Multi-Head attention layer, 768 hidden units, embedding size equal to 300, dropout = 0.2 and batch size equal to 128. Training lasted 3 epochs, about 40 seconds each.

2.5 System 2: Depth-wise Separable Convolutional Neural Network

Depth-wise Separable Convolution (DSC) is a well known technique in Computer Vision to lower dramatically the number of parameters in CNN. DSC consists in decomposing classical 3D convolution, performing at first a depth-wise spatial convolution for each channel, followed by a point-wise convolution which mixes together the resulting output channels. This computational trick achieves in mimicking the true convolution kernel operation, while reducing the size of the model, and speeding up the training with almost the same accuracy.
Our neural network architecture is reported in Figure 3, and takes inspiration from Yoon Kim’s well-known architecture (Kim, 2014). We made some changes taking into consideration both the vectorized text and its PoS Tagging. The overall architecture is made by two parallel DSC networks that receive the text, and PoS embedding respectively. The two convolutional blocks are then averaged through max pooling. After testing various combinations of parameters, we found that the most efficient setup for this task: \([16, 32, 64]\) convolutional filters, kernel size = 2, dropout = 0.3, and batch size = 32. Training lasted 8 epochs, about 5 seconds each.

3 Results

In this Section we describe the HaSpeeDe 2 tasks of the EVALITA 2020 competition, and we present our results obtained in each of them. To evaluate the degree of generality of our approach, we also tested it on hate speech detection tasks for languages other than Italian, that is English, Spanish and German. The official ranking reported for each run is given in terms of macro-average F-score.

3.1 HaSpeeDe 2 Task A - Hate Speech Detection

This is the main task, and it consists of a binary classification aimed at determining whether the message contains Hate Speech or not. We fine-tuned the parameters for this task and then we used the model as it is for the other tasks. We were provided with a labeled training set – made of tweets only – and two unlabeled test sets: one containing in-domain data, i.e. tweets, and the other out-of-domain data, i.e. news headlines. Our results for both Task A test sets are reported in Table 2.

<table>
<thead>
<tr>
<th>Test data</th>
<th>Model</th>
<th>Rank</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>news</td>
<td>Transformer</td>
<td>1/27</td>
<td>0.7744</td>
</tr>
<tr>
<td>news</td>
<td>DSC</td>
<td>4/27</td>
<td>0.7183</td>
</tr>
<tr>
<td>tweets</td>
<td>Transformer</td>
<td>3/27</td>
<td>0.7893</td>
</tr>
<tr>
<td>tweets</td>
<td>DSC</td>
<td>5/27</td>
<td>0.7782</td>
</tr>
</tbody>
</table>

Table 2: Results of the HaSpeeDe 2 Task A

3.2 HaSpeeDe 2 Task B - Stereotype Detection

Task B is a binary classification aimed at determining whether the message contains stereotypes or not. The task is motivated by the fact that stereotypes constitute a common source of error in HS identification (Francesconi et al., 2019). Task B data sets are the same as Task A. Our results for both the in-domain and out-of-domain test sets are reported in Table 3.

<table>
<thead>
<tr>
<th>Test data</th>
<th>Model</th>
<th>Rank</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>news</td>
<td>Transformer</td>
<td>1/12</td>
<td>0.7203</td>
</tr>
<tr>
<td>news</td>
<td>DSC</td>
<td>2/12</td>
<td>0.7184</td>
</tr>
<tr>
<td>tweets</td>
<td>Transformer</td>
<td>3/12</td>
<td>0.7615</td>
</tr>
<tr>
<td>tweets</td>
<td>DSC</td>
<td>5/12</td>
<td>0.7386</td>
</tr>
</tbody>
</table>

Table 3: Results of the HaSpeeDe 2 Task B

3.3 Multilingual Detection of Hate Speech

We tested our systems also against data sets coming from either Hate Speech or Offensive Language detection tasks for other languages.

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.3500</td>
<td>0.4930</td>
</tr>
<tr>
<td>Mean</td>
<td>0.4484</td>
<td>0.6821</td>
</tr>
<tr>
<td>Median</td>
<td>0.4500</td>
<td>0.7010</td>
</tr>
<tr>
<td>Max</td>
<td>0.6510</td>
<td>0.7300</td>
</tr>
</tbody>
</table>

Table 4: Results of the HatEval Subtask A

<table>
<thead>
<tr>
<th></th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.5487</td>
</tr>
<tr>
<td>Mean</td>
<td>0.7151</td>
</tr>
<tr>
<td>Median</td>
<td>0.7295</td>
</tr>
<tr>
<td>Max</td>
<td>0.7695</td>
</tr>
</tbody>
</table>

Table 5: Results of the GermEval 2019 Task 2
4 Discussion

As it can be seen in the results, the Transformer model has always outperformed the DSC model: we expected this outcome due to the nature of the DSC model, designed to be as light as possible but still performing. Regarding the results obtained with the Italian language, we are satisfied with our implementations which have achieved excellent ranking positions in all tasks. In particular, the Transformer model outperformed all the systems that participated to the tasks ranking first with out-of-domain data. This can be seen as a great ability of our model to generalize starting from a training data set different from that of the application. Regarding the results obtained with in-domain data we performed slightly worse, ranking third. This is probably due to the PoS Tagging model that we used in fact it is a model trained on Wikipedia and not on social language, even if slightly modified to manage hashtags, emoticons and URLs, it certainly does not perform well on social texts as if it were a purely PoS Tagging model trained on social media language.

As regards the results obtained with the other languages, we can see that with the Spanish language we get an excellent result, surpassing the first official ranked of the HatEval 2019 competition in Spanish. Our models do not achieve as good results as that of English and German even if the Transformer’s score is always above the median value. We think that this is caused by the nature of languages, because Germanic languages, such as English and German, probably benefit less than Latin ones from the additional use of the PoS Tagging, in the way we used it. We are still investigating how to get added value from PoS Tagging for the English and German languages.

5 Conclusion

In this paper we have introduced two systems for the hate speech detection of social media texts in Italian, Spanish, English and German language. The main feature of these models is to use as input to the neural network not only the pre-processed text, but also it’s PoS Tag. We are satisfied with the results obtained, because the systems implemented are light and performing. Furthermore we have shown that the use of models that include the additional use of the PoS Tagging, to give it more meaning, has given an added value, reached the top positions in the tasks ranking. Our future work will focus on injecting more and more the grammatical structure of a sentence into a model, in fact we are planning a language model that does not only have the purpose of predicting a word based on the given context but that it is also capable of predicting the PoS Tag of that word.

References

Valerio Basile and Malvina Nissim. 2013. Sentiment analysis on Italian tweets. In Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 100–107, Atlanta.


