Hate Speech Detection using Attention-based LSTM

Gretel Liz De la Peña Sarracén[1], Reynaldo Gil Pons[2], Carlos Enrique Muñiz Cuza[2], Paolo Rosso[1]
[1]PRHLT Research Center, Universitat Politècnica de València, Spain
gredela@posgrado.upv.es
prosso@dsic.upv.es
[2]CERPAMID, Cuba
{rey,carlos}@cerpamid.co.cu

Abstract

English. This paper describes the system we developed for EVALITA 2018, the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian, on Hate Speech Detection (HaSpeeDe). The task consists in automatically annotating Italian messages from two popular micro-blogging platforms, Twitter and Facebook, with a boolean value indicating the presence or not of hate speech. We propose an Attention-based in Long Short-Term Memory Recurrent Neural Network where the attention layer helps to calculate the contribution of each part of the text towards targeted hateful messages.

Italiano. *In questo articolo descriviamo il sistema che abbiamo sviluppato per il task di Hate Speech Detection (HaSpeeDe), presso EVALITA 2018, la sesta campagna di valutazione dellelaborazione del linguaggio naturale. Il task consiste nellannotare automaticamente testi italiani da due popolari piattaforme di micro-blogging, Twitter e Facebook, con un valore booleano indicando la presenza o meno di incitamento allodio. Il nostro approccio usa una rete neurale ricorrente LSTM attention-based, in cui il layer di attenzione aiuta a calcolare il contributo di ciascuna porzione del testo verso messaggi di odio mirati.*

1 Introduction

In recent years, Hate Speech (HS) has become a major issue as a hot topic in the domain of social media. Some key aspects (such as virality, or presumed anonymity) that characterize it, distinguish it from offline communication and make it potentially more dangerous and hurtful. Therefore, the identification of HS is an important step for dealing with the urgent need for effective counter measures to this issue.

The evaluation campaign EVALITA 2018[1] launched this year the HaSpeeDe (Hate Speech Detection) task[2] (Bosco et al., 2018). It consists in automatically annotating messages from two popular micro-blogging platforms, Twitter and Facebook, with a boolean value indicating the presence (or not) of HS.

Deep neural network are greatly studied due to their flexibility in capturing nonlinear relationships. Long Short-Term Memory units (LSTM) (Hochreiter and Schmidhuber, 1997) are one of the most used in Natural Language Processing (NLP). They are able to learn the dependencies in lengths of considerably large chains. Moreover, attention models have become an effective mechanism to obtain better results (Yang et al., 2017; Zhang et al., 2017; Wang et al., 2016; Lin et al., 2017; Rush et al., 2015). In (Yang et al., 2016), the authors use a hierarchical attention network for document classification. The model has two levels of attention mechanisms applied at the word and sentence-level, enabling it to attend differentially to more and less important content when constructing the document representation. The experiments show that the architecture outperforms previous methods by a substantial margin. In this paper, we propose a similar Attention-based LSTM for HaSpeeDe. The attention layer is applied on the top of a Bidirectional LSTM to generate a context vector for each word embedding which is then fed to another LSTM network to detect the presence or not of hate in the text. The paper is organized as follows. Section 2 describes our system.

[1] http://www.evalita.it/2018

[2] http://www.di.unito.it/tutreeb/haspeede-evalita18/index.html

Experimental results are then discussed in Section 3. Finally, we present our conclusions with a summary of our findings in Section 4.

2 System

2.1 Preprocessing

In the preprocessing step, the text is cleaned. Firstly, the emoticons are recognized and replaced by corresponding words that express the sentiment they convey. Also, all links and urls are removed. Afterwards, text is morphologically analyzed by FreeLing (Padró and Stanilovsky, 2012). In this way, for each resulting token, its lemma is assigned. Then, the texts are represented as vectors with a word embedding model. We used pretrained word vectors in Italian from fastText (Bojanowski et al., 2016).

2.2 Method

We propose a model that consists in a Bidirectional LSTM neural network (Bi-LSTM) at the word level as Figure 1 shows. At each time step t the Bi-LSTM gets as input a word vector x_t with syntactic and semantic information, known as word embedding (Mikolov et al., 2013). Afterward, an attention layer is applied over each hidden state $\hat{h_t}$. The attention weights are learned using the concatenation of the current hidden state h_t of the Bi-LSTM and the past hidden state s_{t-1} of the Post-Attention LSTM (Pos-Att-LSTM). Finally, the presence of hate (or not) in a text is predicted by this final Pos-Att-LSTM network.

2.3 Bidirectional LSTM

In NLP problems, standard LSTM receives sequentially (left to right order) at each time step a word embedding x_t and produces a hidden state h_t. Each hidden state h_t is calculated as follow:

$$\begin{aligned}
input\_gate_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}) \\
forget\_gate_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}) \\
output\_gate_t &= \sigma(W^{(o)}x_t + U^{(i)}h_{t-1} + b^{(o)}) \\
new\_mem_t &= \sigma(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}) \\
final\_mem_t &= i_t \otimes u_t + f_t \otimes c_{t-1} \\
h_t &= o_t \otimes tanh(c_t)
\end{aligned}$$

Where all W_*, U_* and b_* are parameters to be learned during training. The function σ is the sigmoid function and $\otimes$ stands for element-wise multiplication.

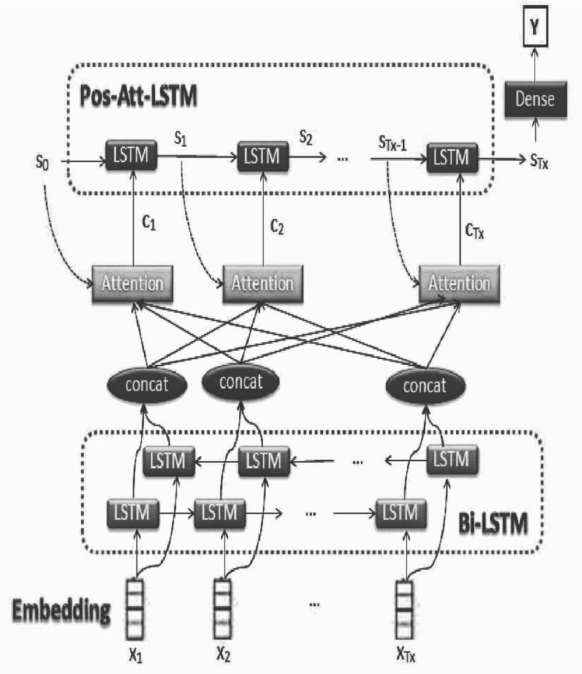

Figure 1: General architecture

The bidirectional LSTM, on the other hand, makes the same operations as standard LSTM but, processes the incoming text in a left-to-right and a right-to-left order in parallel. Thus, the output is a two hidden state at each time step $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$.

The proposed method uses a Bidirectional LSTM network which considers each new hidden state as the concatenation of these two $\hat{h_t} = [\overrightarrow{h_t}, \overleftarrow{h_t}]$. The idea of this Bi-LSTM is to capture long-range and backwards dependencies.

2.4 Attention Layer

With an attention mechanism we allow the Bi-LSTM to decide which part of the sentence should "attend". Importantly, we let the model learn what to attend on the basis of the input sentence and what it has produced so far. Figure 2 shows the general attention mechanism.

Let $H \in R^{2*N_h \times T_x}$ the matrix of hidden states $[\hat{h_1}, \hat{h_2}, ..., \hat{h_{T_x}}]$ produced by the Bi-LSTM, where N_h is the size of the hidden state and T_x is the length of the given sentence. The goal is then to derive a context vector c_t that captures relevant information and feeds it as an input to the next level (Pos-Att-LSTM). Each c_t is calculate as follow:

$$c_t = \sum_{t'=1}^{T_x} \alpha_{t,t'} \hat{h_{t'}}$$

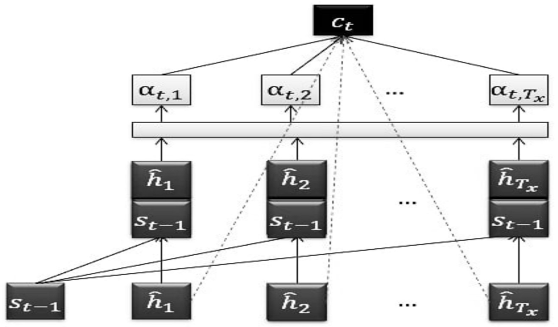

Figure 2: Attention layer

$$\alpha_{t,t'} = \frac{\beta_{t,t'}}{\sum_{i=1}^{T_x} \beta_{t,i}}$$

$$\beta_{t,t'} = tanh(W_a * [\hat{h}_t, s_{t-1}] + b_a)$$

Where W_a and b_a are the trainable attention weights, s_{t-1} is the past hidden state of the Pos-Att-LSTM and $\hat{h}_t$ is the current hidden state. The idea of the concatenation layer is to take into account not only the input sentence but also the past hidden state to produce the attention weights.

2.5 Post-Attention LSTM

The goal of the Post-Att-LSTM is to predict whether the text is hateful or not. This network at each time step receives the context vector c_t which is propagated until the final hidden state s_{T_x}. This vector is a high level representation of the text and is used in the final softmax layer as follow:

$$\hat{y} = softmax(W_g * s_{T_x} + b_g)$$

Where W_g and b_g are the parameters for the softmax layer. Finally, cross entropy is used as the loss function, which is defined as:

$$L = -\sum_i y_i * log(\hat{y}_i)$$

y_i is the true classification of the i-th text.

3 Results

Table 1 shows the results obtained by different variants of the proposed method with the 5-fold cross-validation in terms of F1-score, precision and recall on the training set. The models are: M1 - LSTM+Att+LSTM (run1), M2 - LSTM+Att+LSTM (run2), M3 - Bi-LSTM+Att+LSTM (run1) and M4 - Bi-LSTM+Att+LSTM (run2).

	Twitter			Facebook		
	F1	P	R	F1	P	R
SVM	0.748	0.772	0.737	0.780	0.787	0.781
M1	0.869	0.881	0.863	0.865	0.872	0.863
M2	0.865	0.867	0.865	0.894	0.895	0.894
M3	0.853	0.860	0.854	0.864	0.873	0.864
M4	**0.877**	**0.891**	**0.871**	**0.899**	**0.903**	**0.899**

Table 1: 5-fold cross-validation results on the training corpus (Twitter and Facebook) in terms of F1-score (F1), Precision (P) and Recall (R). The best results are in bold. run2 in M2 and M4, identifies models that take dictionaries into account.

As run1 in M1 and M3, we first evaluated the model described before which is compound for the Bi-LSTM, the Attention layer and the LSTM (Bi-LSTM+Att+LSTM). Also, a variation in this model originated a new model for analizing the contribution of the Bi-LSTM layer. Therefore, we substituted the Bi-LSTM for a LSTM (LSTM+Att+LSTM).

Then, we processed the training sets to generate resources that we called the hate words dictionaries. For each train set we generated a dictionary of the most common words in the texts labeled as hateful. Taking into account this dictionaries, we added a linguistic characteristic to texts which defines if it contains a word into the correspondent dictionary. Thus, run 2 of the model is obtained considering this linguistic characteristic.

We used a SVM as baseline to compare the results of the different variants of the model and all variants achieved better results than this baseline.

The results show that the original model outperforms the results of the variant where the Bi-LSTM is not used. It is important to note that this occurs for run2 where the linguistic characteristic is taken into account. In fact, when this feature is not used the results decrease and the original model obtains the worst results in most cases. Therefore, taking into account the run2 of each variant, the results suggest that the best option is to use the Bi-LSTM with the linguistic characteristic.

The HaSpeeDe task was three sub-tasks, based on the dataset used. First, only the Facebook dataset could be used to classify the Facebook test set (HaSpeeDe-FB), where our system takes macro-average F1-score of 0.7147 and 0.7144, reaching the 11th and 10th positions for run1 and run2 of the model respectively. Another subtask was HaSpeeDe-TW, here only the Twitter dataset can be used to classify the Twitter test set, where our system takes scores of 0.6638 and 0.6567, reaching the 12th and 13th positions for run1 and run2 of the model respectively. Finally, two other tasks consisted of using one of the datasets to train and the other to classify (Cross-HaSpeeDe). Here our system takes scores of 0.4544 and 0.5436, reaching places 10th and 7th in Cross-HaSpeeDe-FB and scores of 0.4451 and 0.318, for places 10th and 12th in Cross-HaSpeeDe-TW.

We think that these results can be improved with a more careful tunning of the model parameters. In addition, it may be necessary to enrich the system with linguistic resources for the treatment of the Italian language.

4 Conclusion

We propose an Attention-based Long Short-Term Memory Network Recurrent Neural Network for the EVALITA 2018 task on Hate Speech Detection (HaSpeeDe). The model consists of a bidirectional LSTM neural network with an attention mechanism that allows to estimate the importance of each word and then, this context vector is used with another LSTM model to estimate whether a text is hateful or not. The results showed that the use of a linguistic characteristic based on the occurrence of hateful words in the texts allows to improve the performance of the model. In addition, experiments performed on the training sets with 5-fold cross-validation suggest that the use of the Bi-LSTM layer is important when this linguistic characteristic is taken into account.

Acknowledgments

The work of the fourth author was partially supported by the SomEMBED TIN2015-71147-C2-1-P research project (MINECO/FEDER).

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Cristina Bosco, Felice Dell'Orletta, Fabio Poletto, Manuela Sanguinetti, and Maurizio Tesconi. 2018. Overview of the Evalita 2018 Hate Speech Detection Task. In Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, *Proceedings of Sixth Evaluation Campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2018)*, Turin, Italy. CEUR.org.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kai Lin, Dazhen Lin, and Donglin Cao. 2017. Sentiment analysis model based on structure attention mechanism. In *UK Workshop on Computational Intelligence*, pages 17–27. Springer.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Lluís Padró and Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. In *LREC2012*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based lstm for aspect-level aentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Min Yang, Wenting Tu, Jingxuan Wang, Fei Xu, and Xiaojun Chen. 2017. Attention based lstm for target dependent sentiment classification. In *AAAI*, pages 5013–5014.

Yu Zhang, Pengyuan Zhang, and Yonghong Yan. 2017. Attention-based lstm with multi-task learning for distant speech recognition. *Proc. Interspeech 2017*, pages 3857–3861.