SINAI-DL at SemEval-2019 Task 5: Recurrent networks and data augmentation by paraphrasing

Arturo Montejo-Ráez, Salud María Jiménez-Zafra,
Miguel Ángel García-Cumbreras, Manuel Carlos Díaz-Galiano
CEATIC / Universidad de Jaén
Las Lagunillas s/n
23071 Jaén (Spain)
{amontejo, sjzafra, magc, mcdiaz}@ujaen.es

Abstract

This paper describes the participation of the SINAI-DL team at Task 5 in SemEval 2019, called HatEval. We have applied some classic neural network layers, like word embeddings and LSTM, to build a neural classifier for both proposed tasks. Due to the small amount of training data provided compared to what is expected for an adequate learning stage in deep architectures, we explore the use of paraphrasing tools as source for data augmentation. Our results show that this method is promising, as some improvement has been found over non-augmented training sets.

1 Introduction

We have participated in SemEval 2019 Task 5, named *HatEval* (Basile et al., 2019), which encourage participants to identify hate speech in tweets. The small amount of training data provided makes difficult to train a deep architecture, so strategies like transfer learning and data augmentation are explored in our work. A trained model for word embeddings in the two languages targeted by the tasks have been considered as transfer learning approach. Paraphrasing the tweets has also been tested for data augmentation, doubling the number of tweets available for training the network.

Our results are promising for English, but no improvements have been found for Spanish. Further analysis on the results and the quality of the paraphrasing tools used is needed, but the scores obtained in English encourage us to consider paraphrasing as a promising help in deep learning for natural language processing.

The paper is organized as follows: Section 2 introduces the two main strategies used to train the neural network: data augmentation and transfer learning. In Section 3, task data is analyzed in order to define hyperparameters values. Section 4 describes the neural network architecture applied. Section 6 gives more details on the paraphrasing approach used to generate more training data. Experiments and results are given in Section 7. Finally, Section 8 closes the contribution with some conclusions and proposals for future work.

2 Data augmentation and transfer learning

Nowadays, deep neural architectures are populating the scientific playground in many scenarios: image recognition speech recognition (Graves et al., 2013) and synthesis (Ze et al., 2013), and, of course, text classification (Zhang et al., 2015). But these supervised learning algorithms demands for valid use different requirements that sometimes are difficult to meet. One of the most difficult to overcame in some cases is the need for a large and varied learning dataset. When there is lack of data, two main strategies can be followed: *transfer learning* and *data augmentation*.

- Transfer learning. This approach proposes to train the network on different task that is learned over a large set of available data of similar nature. For example, train a language model over a Wikipedia dump. Then, last layers can be replaced by those that fit the target task, for instance, sentiment analysis, and then trained on the limited dataset for that task.

- Data augmentation. This approach is commonly used in image recognition, by applying different transformations over training images (rotation, shearing, blurring, mirroring...). In this way, we can augment the size of the training data in several orders of magnitude, making the learning process more feasible and robust.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 480–483
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

We will use both of them in our system for hate speech detection.

3 Data analysis

Organizers provided data consisting of a set of tweets annotated as hateful (1) or not hateful (0) -*HS* label-, with a person (1) or group (0) as target -*TR* label- and as aggresive (1) or not aggresive (0) -*AG* label-. The distribution of tweets per language and dataset is shown in Table 1.

Language	Train	Dev	Test	Total
EN	9,000	1,000	3,000	13,000
ES	4,500	500	1,600	6,600

Table 1: Distribution of tweets per dataset.

We analyzed training and development data to see whether the distribution of labels were similar, obtaining a positive result, which is desirable to guarantee the validity of the model fit. Figure 1 presents ring charts corresponding to the relative values of labels distribution across samples and datasets. Labels are represented using a binary codification of *HS*, *TR* and *AG* annotations. For example, 111 codification corresponds to hateful tweets towards a person and with aggressive content.

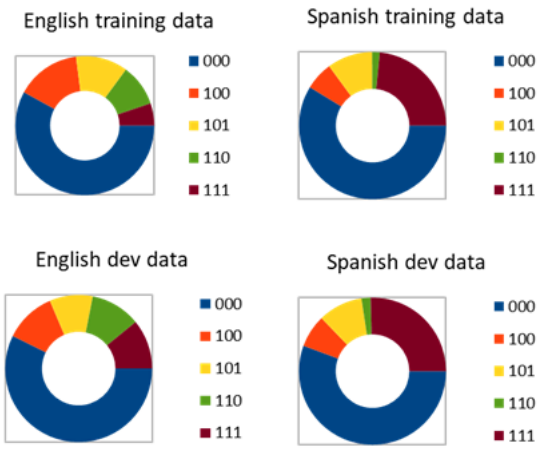

Figure 1: Datasets distribution per label.

In addition, we also analyzed the length of the tweets to decide which window size to use in our experiments. For this, a cumulative histogram for each dataset was generated according to different tweet lengths in order to select a size that would cover a high rate of tweets.

The sizes that cover 80 % and 90 % of tweets are summarize in Table 2 as quantiles 0.8 and 0.9 respectively. A value of 44 for quantile 0.8 means that 80 % of the tweets have a length of 44 or less. Taking into consideration the results we decided to select a window size of 40 words.

Data	Quantile 0.8	Quantile 0.9
train_EN	35	45
dev_EN	44	51
train_ES	37	46
dev_ES	38	49

Table 2: Length of tweets covering 80 % and 90 % of cases.

4 System description

We have implemented the proposed neural network using the Keras[1] library for Python, running on TensorFlow over a NVIDIA Titan X card. Each model took approximately 25 minutes to get trained and few seconds to classify development or test sets. The architecture of our neural network follows a sequence of layers as follows:

1. First layer: An embedding layer that is loaded with pre-trained weights, and converts each word into a 200-dimensional vector for English or a 300-dimensional one for Spanish.

2. Second layer: A bi-directional LSTM recurrent network with 512 activations and a dropout value of 0.5.

3. Third layer: A dense network with 128 activations and the *ReLU* function as activation function. A dropout of 0.5 is also applied after this network.

4. Fourth layer: last classification layer, with 3 activations on the sigmoid function, as we are in a multi-label classification task.

The model has been trained with the hyperparameters values specified in Table 3.

The text have been preprocessed as follows:

1. Lower case is applied.

2. Hashtags are splitted into several tokens according to a camel case approach. For example, "*#MeToo*" is converted into the terms "*<BOH>me too <EOH>*".

[1] http://keras.io

Parameter	value
Batch size	512
Loss function	binary cross-entropy
Optimization algorithm	Adam
Sequence length	40 terms
No. Epochs	100

Table 3: Hyperparameters.

3. Mentions are replaced by the token *<MENTION>*.

4. Unknown terms (those not found in the embedding dictionary) are replaced by the token *<UNK>*.

5. A final token *<EOT>* is added at the end of the tweet.

5 Transfer learning

We have taken already trained word embeddings for the first layer, allowing the weights of the these foreign models to get retrained during the learning process. We have used pre-trained GloVe (Global Vectors for Word Representation) models (Pennington et al., 2014) for the to targeted languages, English[2] and Spanish[3]. These are the models of word embeddings that we have transferred to the first layer in our architecture:

- For English we have used the weights from the GloVe Twitter model provided by the Stanford NLP Group, which is built over 2 billion tweets (27B tokens, 1.2M vocab, uncased, 200-dimensional vectors, 1.42 GB download).

- For For Spanish we have downloaded the GloVe model of the Spanish Billion Word Corpus(Cardellino, 2016), which generated 855,380 vectors of 300 dimensions.

Although FastText (Bojanowski et al., 2017) is considered the state-of-the-art for word embeddings representation, as it considers character n-grams instead of whole word forms, we have opted for GloVe due to the amount of available pretrained models.

[2] https://nlp.stanford.edu/projects/glove/

[3] https://github.com/uchile-nlp/spanish-word-embeddings

6 Data augmentation

The main problem with the dataset mentioned in the previous section is that there is a strong class imbalance between the samples with labels *"000"* and the samples with a different labeling. As Figure 1 illustrates, most of the samples were labeled as not hateful tweets towards a group and with not aggressive content, meaning that this class is highly dominated. Class imbalance introduces two key limitations: firstly, significant differences between accuracy and recall for some classes; and secondly, many machine learning models are prone to overfit on the majority class.

There are a number of ways to counter class imbalance, such as down-sampling the majority class, up-sampling the minority, and other hybrid solutions.

For each tweet, our system expand the information using paraphrasing. To express the same message with different words, we applied a online tool like RewriterTools[4]. For instance, the paraphrase of the tweet *"EU's hailed migrant plan 'a road to Hell' Czech Republic refuses involvement"* was *"EU's hailed migrant layout 'a avenue to Hell' Czech Republic refuses involvement"*.

Different configurations were created with the test data and the system, with the aim of obtaining results and analyzing the behavior of the different modules.

7 Experiments and results

We have performed several experiments to find good hyperparameters, but also evaluated the two main strategies proposed in our approach: transfer learning and data augmentation. In order to verify how transfer learning is good enough, i.e. how the predefined weights for GloVe could be further adjusted or not, we have checked the performance of the model trained on the official training set and evaluated on the development dataset on. Results in Table 4 shows that a small but consistent improvement is obtained if weights can be readjusted.

Next, we have empirically evaluated how paraphrasing helps to produce a better model or not. On these experiments the embeddings are always trainable. The paraphrasing tools allowed us to double the number of tweets in the training dataset. Thus, for English we have 18,000 tweets and

[4] https://www.rewritertools.com/paraphrasing-tool

train	eval	embeddings	F-Score
train_EN	dev_EN	fixed	0.697875
train_EN	dev_EN	trainable	0.707153
train_ES	dev_ES	fixed	0.770951
train_ES	dev_ES	trainable	0.781350

Table 4: Experiments on fixed/trainable embeddings weights.

train	eval	F-Score
train_EN	dev_EN	0.707153
train_aug_EN	dev_EN	0.715593
train_ES	dev_ES	0.781350
train_aug_ES	dev_ES	0.767844

Table 5: Experiments on data augmentation

9,000 for Spanish. Table 5 shows the results obtained. Here different effects are noticiable. For English a slight improvement on macro F-Score metric is reported, and for Spanish the effect is very negative.

We have submitted predictions on the test set on models trained only on task B, so for task A we have submitted only predicted labels for HS column. For English, the training data has been the augmented official training set with paraphrased tweets. For Spanish, only the training tweets provided by the organizers have been used to produce the model. The official results obtained in this task are shown in Table 6

8 Conclusions and future work

Our proposal explores how transferred embeddings and data augmentation may help in a text classification task like HatEval. Paraphrashing does not report clear benefits. This can be due to the quality of the paraphrasing and the fact that new generated tweets are not very realistic. Other augmentation strategies could be explored, like forward-backward translation. We have found also the models trained exhibits high variance. That means that we are overfitting the model on training data, so despite the use of the dropout technique, early stopping, fewer parameters or more training data could help to produce a more robuts model. Another possible improvement is on how final labels are decided. Our system takes the final outputs of the last sigmoid layer as probabilities, so when the value is higher than 0.5 for a class, then the label is 1, 0 otherwise. We could try to set the thresholds using a SVM classifier on this sigmoid vector.

Subtask	F1 (avg)	EMR	rank/total
EN_A	0.519	-	5/70
EN_B	-	0.384	7/41
ES_A	0.686	-	26/39
ES_B	-	0.583	17/23

Table 6: Official HatEval results for our submissions.

Acknowledgements

This research work is partially supported by a grant from the Ministerio de Educación Cultura y Deporte (MECD - scholarship FPU014/00983), the project REDES (TIN2015-65136-C2-1-R) and a grant from the Fondo Europeo de Desarrollo Regional (FEDER).

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)*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Cristian Cardellino. 2016. Spanish Billion Words Corpus and Embeddings.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Heiga Ze, Andrew Senior, and Mike Schuster. 2013. Statistical parametric speech synthesis using deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7962–7966. IEEE.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.