Saagie at Semeval-2019 Task 5: From Universal Text Embeddings and Classical Features to Domain-specific Text Classification

Miriam Benballa
Saagie / LITIS
Rouen, France
miriam.benballa@saagie.com

Sebastien Collet
Saagie
Rouen, France
sebastien.collet@saagie.com

Romain Picot-Clemente
Saagie
Rouen, France
romain.picotclemente@saagie.com

Abstract

This paper describes our contribution to SemEval 2019 Task 5: Hateval. We propose to investigate how domain-specific text classification task can benefit from pretrained state of the art language models and how they can be combined with classical handcrafted features. For this purpose, we propose an approach based on a feature-level Meta-Embedding to let the model choose which features to keep and how to use them.

1 Introduction

In this paper, we describe our system for Task 5 of SemEval 2019 (Basile et al., 2019), namely Multilingual detection of hate speech against immigrants and women in Twitter (HatEval). In this task, participants were asked to automatically classify English and Spanish tweets as hateful or not for Subtask A, and to predict if these tweets are aggressive or not, then identify whether the target is generic or individual for Subtask B. We participated in all subtasks for both English and Spanish.

Our main interest in this competition is to evaluate how a domain-specific dataset can take advantage of unsupervised data and moreover, how very different features can be combined efficiently in a deep neural network to improve classification. For this purpose, we propose to exploit state of the art pretrained deep learning models in text classification and classical features into an architecture that allows combining them dynamically.

Our work consists of three steps: features creation, dynamic meta-embedding and finally combining this information to classify tweets. The next sections are organized as follows: in section 2, we will briefly cover the related work, in section 3 we will explain our model, then in section 4 we will expose our experiences, and finally we will introduce our results in section 5.

2 Related Work

A successful classical approach for tweets classification and sentiment analysis is to use neural networks on top of pre-trained word embeddings. Word embeddings are trained with unsupervised data with a method called distant supervision (Go et al., 2009). Deriu et al. (2016) use Convolutional neural networks on top of those word embeddings while Cliche (2017) is using an ensemble of CNNS and LSTMs. Both solutions won respectively SemEval task 4 in 2016 and 2017.

For tasks more closely related to SemEval Task 5, Sánchez Gómez (2018) won the IberEval 2018 Aggressiveness detection task with an Ensembling of several SVMs models. The Ensembling is done with a Genetic Algorithm. Cuza et al. (2018) propose a model with a Bi-LSTMs network with attention layers on top of pre-trained word embeddings. Their solution got the 2nd place.

On the Mysogyny detection task in IberEval 2018, Pamungkas et al. (2018) won with an SVM trained on a lot of handcrafted features. They used stylistic, structural and lexical features to represent information such as Hashtag Presence, Link Presence, Swear Word Count, Swear Word Presence, Sexist Slurs Presence and Woman-related Words Presence. SemEval 2019 Task 5 is a combination of those two IberEval 2018 tasks.

However, a recent trend in Natural Language Processing has been the use of Transfer Learning from universal sentence embedders to tackle text classification tasks such as Hate Speech detection. This approach is particularly useful when little supervised data is accessible.

The main goal of these universal sentence embedding methods is to embed a sentence in a fixed sized vector that encodes as best as possible the sentence semantic and syntactic information. There are various universal sentence embed-

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 469–475
Minneapolis, Minnesota, USA, June 6–7, 2019.

ding approaches such as the Skip-Thought Vectors (Kiros et al., 2015) that adapt the skip-gram Model of the original Word2Vec to the sentence level, or Infersent (Conneau et al., 2017) that uses a model trained in a supervised fashion on a Natural Language Inference Task.

However, the most promising approaches are probably those based on language models. OpenAI (Radford et al., 2018) propose such a solution called GPT based on the Transformer architecture (Vaswani et al., 2017). In their work, a Transformer is trained in a generative unsupervised manner on a Language Modeling task. The model tries to continuously predict the following word of a text given the rest of the text. Another approach, BERT (Devlin et al., 2018) is also based on the Transformer architecture, but the unsupervised learning scheme is a bit different. The idea is to counter the left-right bias that may arise with classical language modeling. During the training phase, the model tries to predict words hidden randomly in the text and it also tries to tell whether two sentences are following each other or not. These models are trained on datasets such as Wikipedia and BooksCorpus (Zhu et al., 2015).

Both approaches give good results on the GLUE benchmark (Wang et al., 2018), which is a language understanding benchmark based on a diverse range of NLU tasks. Models that present high scores on this benchmark should have a good Transfer Learning capability.

Since the emergence of Word Embeddings with the Word2Vec (Mikolov et al., 2013) in 2013, numerous Word Embeddings approaches were developed such as Glove (Pennington et al., 2014), FastText (Bojanowski et al., 2017) or more recently Elmo (Peters et al., 2018). Evaluating the quality of such Word Embedding in a fair manner is a difficult task and these embeddings approaches may perform best in various situations. Dynamic Meta-Embeddings (Kiela et al., 2018) is a sentence representation method that lets a neural network figure out which Word Embedding from an ensemble to use depending on the situation.

3 Model Description

Universal sentence embedding is a way to share knowledge across different tasks. It is particularly helpful in situations with very small training dataset such as SemEval2019 Task 5 (10000 tweets in the training and development set). A pretrained sentence embedding model aims at a general syntactic and semantic understanding of the tweets.

However, the vocabulary and expressions used in this task are really context-specific so it seems necessary to be able to bring some of this specific content into the universal model. Moreover, we argue each sentence representation can potentially bring additional information to the others. Hence, instead of selecting the best sentence representation for our task, we propose to let a model find the best combination of multiple sentence representations with a Dynamic Meta Embedding approach. This latter works as follows.

From a sentence s, we have n sentence embedding types with different length d_i, leading to a set $\{s_i\}_{i=1}^{n} \in \mathbb{R}^{d_i}$.

Similarly to (Kiela et al., 2018), each sentence embedding is projected to a same d'-dimensional space with a learned linear function $s'_i = P_i s_i + b_i$. where $P_i \in \mathbb{R}^{d' \times d_i}$. These projections are then combined with a weighted sum

$$s_i^{final} = \sum_{i=1}^{n} \alpha_i s'_i$$

where $\alpha_i = g(s'_i)$ are scalar weights which depend on projected sentence embeddings s'_i:

$$\alpha_i = g(s'_i) = \phi(a \cdot s'_i + b)$$

where $a \in \mathbb{R}^{d'}$ and $b \in \mathbb{R}$ are learned parameters and ϕ is a softmax function, so that $\sum_{i=1}^{n} \alpha_i = 1$.

All α_i can be seen as importance weights. When averaging them on all the train dataset, they can be exploited to select important features representations.

For embedding sentences, we propose to use state of the art pretrained models: Bert and GPT. Since they are general sentence embeddings, we finetuned them on our specific tasks to get more specific embeddings (we also tried without finetuning them but got very poor classification results).

Beside these sentence embeddings, we created several classical sentence representations. We constructed all the features suggested by Pamungkas et al. (2018) (see the paper for more details) and some extra features as follows:

- ***Language Model Perplexity*** Perplexity score of each tweet according to the language model kenlm[1] ;

[1] https://github.com/kpu/kenlm

- ***BayesianEncodingHashtag*** Probability of hashtag according to the target class ;
- ***hashtagUrlPresence*** One-Hot encoding on presence of urls and hashtags in tweets ;
- ***Abreviation*** Abreviation counting from a custom lexicon ;
- ***BagOfPOSTagging*** Counting the different POS tags in each tweet ;
- ***NMF*** Non-negative Matrix Factorization on the co-occurrence matrix of words ;
- ***LDA*** Latent Dirichlet Allocation on the tweets ;
- ***BagOfEmojiFeatures*** One-Hot encoding on presence of emojis in tweets ;
- ***nbWords*** Number of words in each tweet, normalized by mean ;
- ***Textstat*** Readability features according to the python package textstat[2] ;
- ***nbChar*** Number of characters in each tweet, normalized by mean.

However, the importance weights from the dynamic weighted sum of our model show that most of these representations were not of interest for the predictions, and were reducing the F1-score. Hence, we made a feature selection based on these weights for each subtask. In the next subsections, we detail the different sentence representations we used for each subtask.

3.1 Pre-processing

We didn't use a lot of pre-processing besides lowercasing, in order to benefit from the representations capabilities of BERT and GPT. These models are using BPE encoding (Sennrich et al., 2015), so the models are based on subword units and not on words. This way, out of vocabulary words such as those with spelling mistakes or very context-specific ones may still be processed in a useful way by the model. However, a kind of spelling mistake correction might have been useful. The main pre-processing scheme we used is the replacement of usernames and urls by a specific token.

We normalized the most frequent hashtags in order to keep only one spelling (for instance #buildthatwall and #buildthewall were processed to have the same spelling). We also processed the most frequent abbreviations by replacing them with their full form. Finally we tried a splitting words approach on the hashtags in order to help the BPE encoding to get sensible of subword units. This did not improve performance, so this preprocessing was not kept in our final submission.

3.2 Subtask A en: Hateful or not

This subtask consists in classifying each English tweet as hateful or non hateful. For each tweet, the following features have been selected and given as inputs to our model:

- Bert embeddings: 3 different finetuned pretrained Bert embeddings[3], one for each target class (HS, TR, AG). Leading to 3 sentence representations of 768 features.
- GPT embeddings: 3 different finetuned pretrained GPT embeddings[4], one for each target class (HS, TR, AG). Leading to 3 sentence representations of 768 features.
- Hate Word Count: Count the presence of words into a lexicon extracted from HateBase[5], leading to 1 feature
- Bag of Emojis: Count the presence of Emojis grouped by type, leading to 155 features (number of emojis)

3.3 Subtask B en: Target and Aggressivity

Subtask B consists in predicting in addition to the hate speech, the target of the hate speech (TR - a group or an individual) and the aggressiveness (AG). We used the same approach, architecture and features to predict the labels TR and AG. Each label is predicted independently. However, we added a simple post-processing correction based on the predictions we made for HS: if a tweet is classified as not hateful, we set the target to generic (TR prediction to 0) and labeled the tweet as not aggressive (AG prediction to 0). This rule has been deducted from the way tweets are labeled: non hateful tweets are always classified as generic and not aggressive.

3.4 Subtask A/B es

We used the same model for the Spanish dataset and translated Spanish tweets to English with machine translation. In doing this, we could employ the same type of features as we used for English.

[2] https://github.com/shivam5992/textstat

[3] https://github.com/huggingface/pytorch-pretrained-BERT

[4] https://github.com/huggingface/pytorch-openai-transformer-lm

[5] https://hatebase.org/

Dataset	Total	HS	TR	AG
Train EN	9000	3783	1341	1559
Dev EN	1000	427	219	204
Test EN	2970	1679	522	590
Train ES	4500	1857	108	1001
Dev ES	500	222	137	176
Test ES	1599	660	423	474

Table 1: Hate speech dataset.

For the subtask B, the same corrections were applied for TR and AG using HS predictions.

4 Experimental Setup

4.1 Data

For each language, a training, a development and a test set were provided. These datasets were manually annotated using Figur8[6] crowdsourcing platform. Statistics on label distribution can be found in Table 1.

4.2 Parameter settings

Our model is implemented in PyTorch[7] and trained on 2 GPU Tesla V100. For the finetuning of Bert and GPT, we used the default parameters of their respective repository but trained on 10 epochs.

For the learning of the Meta Embedding model, we used a batch size of 64 and Adam optimizer with a variable learning rate (the Noam decay introduced in Vaswani et al. (2017)). The dropout rate is set to 0.6. To avoid over-fitting issues and to be able to reproduce and compare our results, we used Scikit-learn[8] implementation of Stratified Shuffle Split, with 10 splits on the concatenated train and dev dataset. Our results metrics are the means of the values obtained on the 10 splits.

5 Results

This section presents the evaluation of the SemEval-2019 Task 5: HatEval. The official measure for this task was the macro F1-measure. Note that for the Subtask B, evaluation was based on two criteria (each dimension evaluated independently or jointly), however the final ranking was based solely on the second criteria (Exact Match Ratio on the three labels). More details about the evaluation system can be found in the task description paper (Basile et al., 2019).

We saved the best epoch model for each of the 10 splits and we used them to make our final prediction for the test dataset. Then we used our 10 models to classify each tweet: to predict a tweet as hateful, at least half (5) of the models have to agree with this class. The same goes for subtask B to predict TR and AG, with in addition the post-processing described in subsection 3.3. Macro F1-scores and EMR scores with this agreement rule on the English development splits and test datasets are respectively presented in Table 2 and Table 3. This latter is our final submission for the competition. We can see a surprising decreasing of macro F1-score for the predictions on the test dataset of about 35 points compared to the predictions on our experimentation splits. We discuss about this result in the next section.

Table 4 and Table 5 show the results on the Spanish datasets. We can see that the finetuned BERT model gives good results on the test dataset (3 points better in macro F1 than the leader on subtask A) whereas it was worse than the other models on the splits dataset on our experiments. Our hypothesis is that the other models may have overfitted on the train dataset (especially the GPT model). Our Meta-embedding model seems to have been penalized by the GPT overfitting.

6 Discussion

6.1 Unsuccessful approaches

During this competition, we experimented many additional methods that did not successfully improve the results:

- We created an important quantity of features manually as described in section 3. However, most of them were not useful for the prediction according to the weights extracted from our model. We suppose this is either because the features from finetuned BERT and GPT models are able to capture most of the information provided by the other features or because it is difficult to blend handcrafted features with the ones obtained from BERT and GPT.

- We tried others universal sentence embedding models besides GPT and BERT such as InferSent and ULMFiT (Howard and Ruder, 2018) but without very good results. As

[6] https://www.figure-eight.com/
[7] https://pytorch.org/
[8] https://scikit-learn.org/stable/

Model	HS (subtask A)	TR	AG	EMR (subtask B)
GPT	82.33%	82.27%	71.10%	66.74%
BERT	82.87%	82.88%	74.11%	68.61%
Meta-Embedding (our submission)	**84.16%**	**83.93%**	**75.01%**	**70.55%**

Table 2: Mean results on English development splits.

Model	HS (subtask A)	TR	AG	EMR (subtask B)
GPT	**51.50%**	**73.78%**	**60.42%**	36.20%
BERT	48.84%	73.39%	59.16%	36.03%
Meta-Embedding (our submission)	49.60%	72.40%	57.80%	**37.40%**
Baseline SVM	45.00%	69.70%	58.70%	30.80%

Table 3: Final results on English test dataset.

SemEval task 5 is very specific vocabulary-wise, it is possible that universal models such as InferSent and ULMFit were not trained on enough data to provide good features representations of the tweets.

- For the Spanish dataset, we also tried the BERT Multilingual model which was released during the competition, but we also had lower results than using translation.

- We tried augmenting the dataset with external resources, especially with a similar labeled dataset[9] of tweets with hate speech and offensive language. Nevertheless, this method was decreasing the results, probably due to different labeling rules.

- Inspired from the back-translation proposed in (Edunov et al., 2018), we augmented the dataset by automatically translating each tweet to an other language (French, Spanish, Chinese) and back translated it to the initial language (English). This method can be assimilated to a transfer learning approach that should bring more variability in the dataset, and should improve the generalization ability of the model. Our tests did not show relevant improvement in F1-score but were decreasing the variance. Nevertheless, we did not develop enough this approach to conclude on its potential benefits.

- About our post-processing choice for Task B, we based it on the fact that our model for HS prediction was better than the models for TR and AG. Considering how much the performance on the HS task decreased on the test set compared with the decreasing on the TR and AG tasks it was probably not the best choice. A multi-label model might have been useful for this task considering the evaluation metric (each label prediction should not be independent).

- Finally, we tried to train our model on both English and translated Spanish datasets, but that did not improve our results.

6.2 About the testing set

The previous section shows an important difference on HS in terms of prediction quality (F1-score) between the development and the test datasets. This score difference seems to be experienced by every participant according to the development and test leaderboards. It seems that the test dataset contains a lot more of difficult tweets to classify in comparison with the train and development datasets. Our hypothesis is that the test dataset has not been collected like the other datasets (train and development) or that data were sorted in a particular way after the collection, which could explain such results.

In this setup it is interesting to see that the features extracted from the finetuned GPT generalize a little better (with a HS F1-score of 51.50) than our submited model (49.6 HS F1-score). Adding more features might have induced more overfitting on the training set.

Since the end of the competition, the state of the art on Natural Language Understanding on the GLUE Benchmark is a new model. It is a Multi-

[9] https://github.com/t-davidson/hate-speech-and-offensive-language/tree/master/data

Model	HS (subtask A)	TR	AG	EMR (subtask B)
GPT	81.67%	86.17%	79.77%	70.10%
BERT	78.21%	81.63%	75.48%	65.22%
Meta-Embedding (our submission)	**83.19%**	**86.79%**	**81.01%**	**74.98%**

Table 4: Mean results on Spanish development splits.

Model	HS (subtask A)	TR	AG	EMR (subtask B)
GPT	66.42%	74.17%	67.66%	51.53%
BERT	**76.88%**	**81.08%**	**76.55%**	**65.85%**
Meta-Embedding (our submission)	71.70%	80.90%	76.00%	63.50%
Baseline SVM	70.10%	78.10%	72.60%	60.50%

Table 5: Final results on Spanish test dataset.

Task Model based on BERT (Liu et al., 2019). It seems that the Multi-Task Learning approach improves the universality of BERT. We think that such a model could also improve our architecture on this task because a model trained in a Multi-Task manner should in theory be more robust to overfitting.

7 Conclusion

In this work, we investigated how a model could merge features obtained from unsupervised language models such as GPT and BERT with domain specific hand-crafted features. We presented an approach based on a feature-level Meta-Embedding to let the model choose which features to keep and how to use them. Our method systematically outperforms models based only on BERT or GPT features on our evaluation datasets, however it is not always the case on the test datasets. For instance, on the Spanish test dataset, BERT alone gives better results and on the English test dataset subtask A, GPT slightly outperforms our submission.

Our idea was that the data used for SemEval 2019 Task 5 is very domain-specific and present a peculiar vocabulary. We thought that universal sentence embeddings methods would not work very well since such vocabulary was probably not present during their unsupervised training and the sentence quality is also probably different. However, our results tend to show that it is not the case. For instance, a model using only BERT features would have been 1st on the Spanish task A. The BPE used as a pre-processing for these models is probably helping to deal with out-of-vocabulary words. On top of that, it seems that big unsupervised language models are able to learn data representation that generalize really well to unseen domains.

Acknowledgments

This work is part of a French RAPID project called SAPhIRS, in collaboration with LITIS Laboratory and Airbus Defence Space. The authors would like to thank Alexandre Pauchet and Simon Bernard for their help.

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.

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.

Mathieu Cliche. 2017. Bb_twtr at semeval-2017 task 4: twitter sentiment analysis with cnns and lstms. *arXiv preprint arXiv:1704.06125*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Carlos Enrique Muniz Cuza, Gretel Liz De la Pena Sarracén, and Paolo Rosso. 2018. Attention mechanism for aggressive detection. In *CEUR Workshop Proceedings*, volume 2150, pages 114–118.

Jan Deriu, Maurice Gonzenbach, Fatih Uzdilli, Aurelien Lucchi, Valeria De Luca, and Martin Jaggi. 2016. Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *Proceedings of the 10th international workshop on semantic evaluation*, CONF.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12).

Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic meta-embeddings for improved sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1477.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Endang Wahyu Pamungkas, Alessandra Teresa Cignarella, Valerio Basile, Viviana Patti, et al. 2018. 14-exlab@ unito for ami at ibereval2018: Exploiting lexical knowledge for detecting misogyny in english and spanish tweets. In *3rd Workshop on Evaluation of Human Language Technologies for Iberian Languages, IberEval 2018*, volume 2150, pages 234–241. CEUR-WS.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Claudia Nallely Sánchez Gómez. 2018. Ingeotec at mex-a3t: Author profiling and aggressiveness analysis in twitter using μtc and evomsa. *OPENAIRE*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.