GTH-UPM at DETOXIS-IberLEF 2021: Automatic Detection of Toxic Comments in Social Networks

Sergio Esteban Romero, Ricardo Kleinlein, Cristina Luna-Jiménez, Juan Manuel Montero[0000−0002−7908−5400], and Fernando Fernández-Martínez[0000−0003−3877−0089]

Speech Technology Group, Center for Information Processing and Telecommunications, E.T.S.I. de Telecomunicación, Universidad Politécnica de Madrid, Av. Complutense, N° 30, Madrid, 28040, Spain
sergio.estebanro@alumnos.upm.es, {ricardo.kleinlein, cristina.lunaj, juanmanuel.montero, fernando.fernandezm}@upm.es

Abstract. Sadly, the presence of toxic messages on social networks, whether in the form of stereotypes, sarcasm, mockery, insult, inappropriate language, aggressiveness, intolerance, or typical of hate speech against immigrants and / or women, among others, is relatively frequent. This presence should not be ignored by the scientific community, since it is their responsibility to develop tools and systems that allow their automatic detection and elimination. In this paper, we present an exploratory analysis in which different deep learning (DL) models for the detection of toxic expressions have been evaluated on the DETOXIS-IberLEF 2021 challenge using the official release of the NewsCom-TOX corpus. Particularly, we compare traditional RNN and state-of-the-art transformer models. Our experiments confirmed that optimum performance can be obtained from transformer models. Specifically, top performance was achieved by fine tuning a BETO model (the pre-trained BERT model for the Spanish language from the Universidad de Chile) for the toxicity detection tasks. Another contribution of this analysis is the validation of the proposed method for adding task-specific vocabulary (new tokens) that could help to effectively extend the original vocabulary of the pre-trained models.

Keywords: Classification task · Toxicity detection · Recurrent networks · Attention · Transformer models · Transfer learning · Social networks

1 Introduction

The automatic detection of toxic language, especially in online tweets and comments, is a task that has attracted growing interest from the NLP (Natural Lan-

IberLEF 2021, September 2021, Málaga, Spain.

guage Processing) community in recent years and has become a tremendously popular and active research area because of its impact on modern society.

In this regard, the DETOXIS challenge is a great opportunity to tackle the hard task of identifying toxic comments in social media. The detection of toxicity is not an easy task at all. It involves much more than just identifying some specific words or sentences, we must also take the context into account which makes the task even more complex. The present work is well aligned with such interest and its objective is the design and implementation of computational models for toxicity assessment and classification of comments in Spanish. Different models have been proposed and evaluated for both subtasks of the challenge:

- Subtask 1: Toxicity detection task is a binary classification task that consists of classifying the content of a comment as toxic (toxic=yes) or not toxic (toxic=no).
- Subtask 2: Toxicity level detection task is a more fine grained classification task in which the aim is to identify the level of toxicity of a comment (0= not toxic; 1= mildly toxic; 2= toxic and 3: very toxic).

In this paper we will present the different toxicity recognition models we have developed as part of our participation in DETOXIS-Iberlef 2021. Our first model is an extension of the system we previously developed for intent detection and classification based on word-embeddings and recurrent neural networks [6]. Our second model, the model that we have used in our final submissions to the challenge, is a transformer based model which has been developed by fine tuning a BETO model (the pre-trained BERT model for the Spanish language from the Universidad de Chile) for the toxicity detection tasks. The lessons learned from our experiments and experience in participating in the challenge have been reported in the next sections.

2 Related work

Automatic detection of toxic comments to ensure its deletion is mandatory in a world surrounded by social media. Traditional methods such as RNN models have provided efficient solutions for similar tasks in multiple fields. Also, RNN models based on Bi-LSTMs allow to explore the context of the sentence improving its performance. However, recent advances such as the BERT model have greatly contributed to enhance natural language processing [5]. The BERT family of algorithms is based on the Transformer architecture [22], a particular type of neural processing unit that outperforms traditional LSTM cells [9]. Transformers have become state-of-the-art models in many of the most popular NLP tasks including automatic toxicity detection in texts [19][13].

3 The NewsCom-TOX dataset

Data provided for the challenge is grouped in NewsCom-Tox dataset which contains around 4357 posts. Most comments are in response of articles from different

Spanish newspapers. Each comment is classified as toxic or not toxic and also in four different levels of toxicity. In addition, some other features are included such as argumentation, sarcasm, mockery or insult, for instance [21]. Furthermore, we must consider that classes are not balanced since it contains a higher amount of non toxic comments.

4 RNN based model

As a first solution for the toxicity analysis we have used a Recurrent Neural Network (RNN), a type of model widely used in the analysis of Twitter messages, for example. RNNs have the ability to process their inputs sequentially, performing the same operation, $h_t = f_W(x_t, h_{t-1})$, on each of the different elements that constitute our input sequence (i.e. words or, to be more exact, their corresponding embeddings), where h_t is the hidden state, t the time step, and W the weights of the network.

As it can be observed, the operation is formulated in such a way that the hidden state at each time step depends on the previous hidden states. Hence, the order of the elements in our sequences (i.e. the order of the words) is particularly important. As an immediate consequence, RNNs allow us to handle inputs (i.e. sentences) of variable length, which happens to be an essential feature given the nature of our problem.

Among the different possible architectures of this type of networks, we have opted for the so-called Long Short-Term Memory (LSTM) networks [2], a special type of RNNs that help preventing the typical vanishing gradient problem of standard RNNs by introducing a gating mechanism to ensure proper gradient flow through the network. LSTMs main characteristic is the ability to learn long-term dependencies. To do this, these networks are supported by basic constituent units called *cells* that are provided with mechanisms that allow deciding for each cell what information is preserved from that provided by the previous cells, and what information is provided to the next ones, both depending on the cell's current state.

4.1 Embeddings

(Word) embeddings are vector-type representations obtained for words in reduced-dimensional vector spaces where semantically similar words are always close to each other. The Fasttext project [8], recently open-sourced by Facebook Research, enables a fast and effective method to learn word embeddings that are very useful in text classification, clustering and information retrieval. In this work, the proposed model uses Fasttext word embeddings to represent the vectors for the words as input of the network.

At the time of training, FastText trains by sliding a window over the input text and either learning the target word from the remaining context (also known as continuous bag of words, CBOW), or all the context words from the target word ("Skip-gram"). Learning can be viewed as a series of updates to a neural

network with two layers of weights and three layers of neurons, in which the outer layer has one neuron for each word in the vocabulary and the hidden layer has as many neurons as there are dimensions in the embedding space. This approach is very similar to Word2Vec [14]. However, unlike Word2Vec, fastText might also learn vectors for sub-parts of words: so-called character n-grams. This ensures that for instance the words love, loved and beloved all have similar vector representations, even if they tend to show up in different contexts. This feature enhances learning on heavily inflected languages [3].

4.2 Model description

Our approach is based on a 2-layer Bidirectional-LSTM model with a deep self-attention mechanism which is represented in Figure 1. The model is implemented in Pytorch [15] and based on the architecture proposed in [2].

Embedding layer The model is designed to work with sequences of words as inputs, thus allowing us to process any type of sentence. For this, a first embedding layer is provided that collects the embeddings $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ corresponding to each of the words $w_1, w_2, \ldots, w_N$ constituing the sentence we want to process, where N is the number of words in our sentence. We initialize the weights of the embedding layer with our pre-trained word embeddings.

Bi-LSTM layer A standard LSTM model behaves in a unidirectional way, that is, the network takes as input the direct sequence of word embeddings and produces the outputs $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N$, where $\boldsymbol{h}_i$ is the hidden state of the LSTM cell at time step i, summarizing all the information that the network has accumulated from our sentence up to word w_i.

Instead, we have used a *bi-directional LSTM* (Bi-LSTM) that allows us to collect such information in both directions. In particular, a Bi-LSTM consists of 2 LSTMs, a *forward* $\overrightarrow{LSTM}$ that allows the analysis of the sentence from w_1 to w_N, and an *inverse or backward* $\overleftarrow{LSTM}$ which allows a similar analysis to be carried out but in the opposite direction, from w_N to w_1. To obtain the definitive outputs of our Bi-LSTM layer, we simply concatenate for each word the outputs obtained from the analysis performed in each specific direction (see Equation 1 in which $||$ corresponds to the concatenation operator and L to the size of each LSTM).

$$h_i = \overrightarrow{h_i} || \overleftarrow{h_i} \text{ , where } h_i \in R^{2L} \tag{1}$$

Attention layer In order to identify the most informative words when determining the polarity of the sentence, the model uses a deep self-attention mechanism. Thus, actual importance and contribution of each word is estimated by means of a multilayer perceptron (MLP) composed of 2 layers with a non-linear activation function (*tanh*) similar to that proposed in [16].

The MLP learns the attention function g as a probability distribution on the hidden states h_i, that allows us to obtain the attention weights a_i that each word receives. As the output of the attention layer the model simply computes the convex combination r of the LSTM outputs h_i with weights a_i, where a convex combination is a linear combination of points where all the coefficients are non-negative and add up to 1.

Output layer Finally, we use r as a feature vector which we feed to a final task-specific layer for classification. In particular, we use a fully-connected layer, followed by a *softmax* operation, which outputs the probability distribution over the classes.

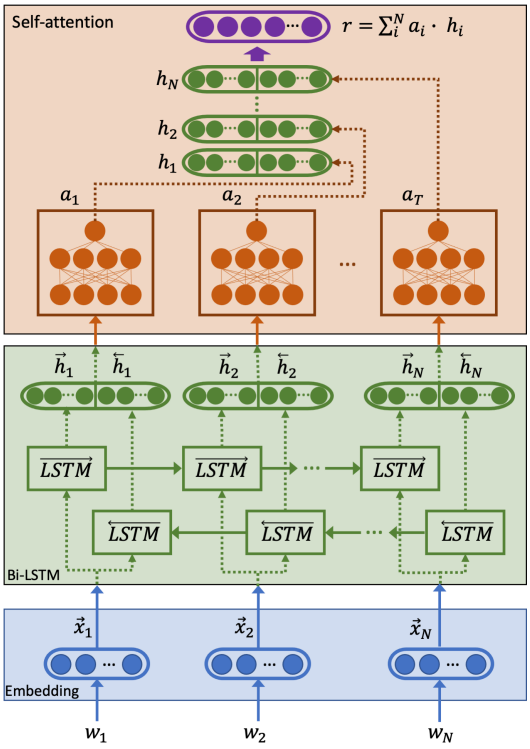

Fig. 1. Proposed RNN based model.

5 Transformer based model

Willing to improve the results obtained with the RNN model, we decided to use transformers, a type of model able to process its inputs sequentially as well. For

this task we used BERT which stands for *Bidirectional Encoder Representations from Transformers* allowing to work in both directions. Also, something that make them so powerful is their attention mechanism for identifying which are the keywords in a sentence. These models usually receive sentences as inputs that are divided into single tokens, obtaining a sequence of them.

The way this process is carried out depends on the tokenizer used but BERT's one is based on words and subwords. So, for instance, if a word is not included in the original vocabulary, it will be divided in a sequence of subtokens that all together would form the initial word. However, when we want to finetune our model within a specific field, it is usual that this occurs with many common words related to the topic, so what we decided to test is whether adding new tokens to our initial vocabulary results in a better toxicity recognition. These new tokens will correspond to the most frequent words in our training dataset that were not already included in the original tokenizer vocabulary.

5.1 Model adaptation

Pre-trained NLP models have led to breakthrough performance improvements in different tasks including intent recognition or sentiment analysis, among many others [17]. However, the adoption of pre-trained language models still must face two important challenges as their applications expand [20]:

1. First, the need for large training resources; training requires substantial computation and data, see, e.g., BERTlarge [5], RoBERTa [12], while most common situation is that available training resources are significantly constrained or limited.
2. Second, the need for extending pre-trained models with domain-specific vocabulary: every target domain, such as the social media domain on which this work focuses, has its own vocabulary, and sentences in the domain may have words from both the original language model's vocabulary and new domain-specific vocabulary. Being able to operate on this mixture of vocabulary is essential in achieving high performance on downstream tasks in the new domain [7].

Instead of constructing our model with a new vocabulary from scratch, which would require substantial computational resources and training data, or simply adapting the existing pre-trained model on the original vocabulary, which would lead to sub-optimal performance on downstream tasks, we have adopted a simple but effective approach that addresses both challenges explicitly. Particularly, our method aims at including only a reduced subset of words from the new domain's vocabulary, carefully selected in a rationale way by attending to their actual frequency in our training data, while being able to reuse and adapt the original pre-trained model. This helps reducing required computation and training data while enhancing recognition performance.

6 Evaluation

6.1 Experimental setup

To prevent overfitting all the experiments have been carried out following a 5-fold cross-validation scheme. Each setup has been trained for 100 epochs and 'Stop-Early' has been adopted as the stopping criterion. No exhaustive exploration of the hyper-parameters of our models was conducted. Models were trained during 100 epochs using an Adam optimizer [11], with initial learning rate of 0.001, batch size of 32, and early-stopping after 5 epochs without improvement in the F1 classification score. For the calculation of F1 we used the *weighted* version that takes into account the number of examples available for each different class.

RNN specific setup With regards to the RNN model both the bi-LSTM and attention layers had a 0.3 dropout rate. The encoder layers had a size L of 150 or 200. As a way to increase input variability from epoch to epoch, input embeddings were randomly added white noise with 0.15 probability rate in order to increase the robustness of the model. Also, given that classes were not perfectly balanced, to prevent introducing bias in our models we applied class weights to the loss function, penalizing more the misclassification of under-represented classes. These weights were computed as the inverse frequencies of the classes in the training set.

BERT specific setup Our BERT model has been implemented and fine-tuned for the toxic comments classification tasks using the Simple Transformers library [18]. Although pre-trained tokenizers work at both word and subword levels, the top N new tokens to be added to the vocabulary (i.e. those that happen to be the most frequent in our training data) have been included as word-level units. The rest of new tokens, those connected to more infrequent words, just get split into smaller units to ensure that there are no out-of-vocabulary tokens and all vocabulary units get updated reasonably frequently during training.

6.2 RNN model results

We have evaluated two different RNN like models whose main difference is the encoder size. Corresponding results are detailed in Table 1. Other values were also tested though size 200 yielded best performance. Nonetheless, performance has demonstrated to be significantly better for even the most simple version of our BERT based model: the model fine-tuned from the cased version of BETO, a BERT model trained on a big Spanish corpus that can be found in [4], without explicitly adding any new domain-specific vocabulary.

6.3 BERT model results

After confirming the superiority of the BERT based approach, we compared different pre-trained models' performance after fine-tuning them on the first

Table 1. RNN vs BERT 5-fold CV results for DETOXIS-Iberlef 2021 Subtask 1.

Model	Description	Weighted F1
RNN1	Encoder size 150	72.17 %
RNN2	Encoder size 200	72.50 %
Cased-BETO	Frozen tokenizer	**75.26 %**

downstream subtask: identifying whether a comment is toxic or not, a binary classification problem. As shown in Figure 2, the cased version of BETO clearly outperforms the other two models: the uncased BETO version and the standard BERT multilingual base model, a model pre-trained on the top 104 languages with the largest Wikipedia using a masked language modeling (MLM) objective [5].

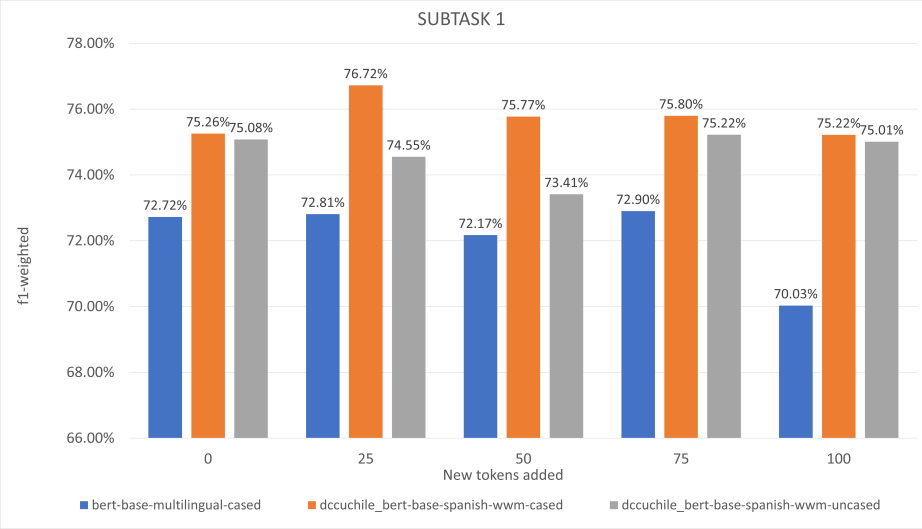

Fig. 2. BERT models comparison: 5-fold CV results for DETOXIS-Iberlef 2021 Subtask 1.

Then, besides comparing alternative pre-trained models, we also measured the impact of our vocabulary extension method. The results of this analysis have been also presented in Figure 2. Results shown there correspond to independent experiments in which a different number of N new tokens (i.e. top N words included as new words) is tested. Results are reported starting from 25 new tokens at first and up to 100, increasing the amount by 25 on each different run.

As it can be deduced from the figure, our vocabulary extension method demonstrates to be effective achieving a top performance of 76.72% and an improvement over the baseline performance obtained when our model is simply fine-tuned without explicitly adding any new word (i.e. 75.26%, previously

reported in Table 1). However, results become worse when the amount of new tokens exceeds a certain small limit, which suggests the importance of finding an adequate balance between the increased complexity of our target model (i.e. the number of new units-embeddings to be learnt) and the available training data.

For the second subtask we have followed exactly the same procedures obtaining similar results as the shown above. In this subtask we are facing a multi-class classification problem, where the goal is to identify the toxicity level of every comment in a 0 to 3 scale (i.e. 0: not toxic; 1: mildly toxic; 2: toxic and 3: very toxic). The evaluation results for the adopted experimental setup based on the 5-fold cross-validation scheme have been presented in Figure 3. In this case, only results obtained for our top-performing approach based on the cased version of the BETO pre-trained model have been reported. Besides, and for a proper comparison and analysis, the result corresponding to the case where we simply fine-tune the model while freezing the original tokenizer (i.e. we do not add any new domain-specific vocabulary) has also been included at the beginning of the series (i.e. the "0" column). Again, we demonstrate that our vocabulary extension method consistently outperforms the prior approach based on general vocabulary. However, once again we confirm that new additive vocabulary can only be introduced to some extent because no further improvement is observed beyond 100 new domain-specific words (best performance is achieved again by the 25 new tokens configuration).

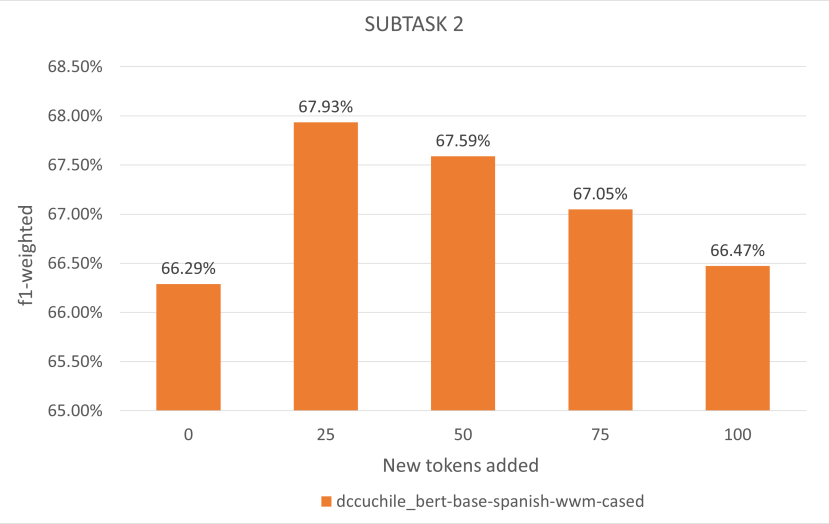

Fig. 3. Analysis of the vocabulary extension method for the top-performing approach based on the cased version of the BETO pre-trained model: 5-fold CV results for DETOXIS-Iberlef 2021 Subtask 2.

Analyzing the effect of different data pre-processing methods Adopting the top-performing approach based on the cased version of the BETO pre-trained model as a reference, we decided to further explore the use of some of the most popular text pre-processing techniques to find out whether they are actually useful or not. Evaluated techniques include the following:

- Removing stop words: stop words are removed with the help of the *spaCy* library [10].
- Removing punctuation: punctuation marks are removed with the help of the Spacy library.
- Lemmatization: Spacy lemmatization is applied to generate the root form of the words.
- Basic text normalization: special words including emojis, emails, percentages, money, phone numbers, times, dates, urls and/or hashtags are assimilated and replaced by a special tokens, such as MAIL, DATE, URL,... to prevent information from being lost during data representation.

Table 2. Results obtained for our top-performing approach based on the cased version of the BETO pre-trained model when enriched with 25 new tokens and applying a specific text preprocessing technique: 5-fold CV results for DETOXIS-Iberlef 2021 Subtask 1.

Technique	Weighted F1
without basic preprocessing	75.91 %
lemmatization	74.11 %
removing stopwords	74.03 %
removing punctuation	73.82 %

The obtained results have been summarised and sorted in terms of performance in Table 2. As it can be observed, none of the applied techniques that are aimed at removing tokens were found to be effective. In short, DL methods that do use embedding representations seems to not require the removal of anything. Specifically, in that n-dimensional vector words like "dogs" and "dog" would already be closer to each other. So, the need to lemmatize becomes unnecessary.

With regards to stopwords, although it could be convenient to remove many of them, we should notice that stopword lists may contain words which shouldn't be removed in certain domains or tasks, as it happens to be the case. Generally speaking, we should not remove anything (e.g., a word or a punctuation mark) that could be useful in some way. Again, DL models working with vector embeddings, similarly to lemmatization, are currently the best methods to handle and filter those irrelevant terms.

Finally, it is interesting to conclude that basic text normalization has proven to be successful (i.e. performance decreases if we omit it). In this case, the process of transforming some words into their single canonical form still helps our model by reducing the number of unique words (i.e. reducing the vocabulary size helps reducing the model complexity and improving its performance).

6.4 DETOXIS-Iberlef 2021 Challenge results

After carefully analyzing the obtained results that have been previously presented, we decided to submit the runs on both DETOXIS-Iberlef 2021 challenge subtasks for our top-performing approach based on the cased version of the BETO pre-trained model when gradually increasing the amount of new tokens in steps of 25 from 0 to 100, thus resulting in 5 different runs and submissions for each task. Details about the challenge and its evaluation are presented in [21].

Table 3. DETOXIS-Iberlef 2021 SUBTASK 1 Top 5 ranking

Ranking	Team Name	F1 Toxic
	Gold_standard	1.000
1	SINAI	0.6461
2	FuillemGSubies	0.600
3	AI-UPV	0.5996
4	DCG	0.5734
5	**GTH-UPM**	**0.5726**

In both tasks, we are named as GTH-UPM. These rankings are available on DETOXIS-Iberlef 2021 official website [1]. Our result in Table 3 corresponds to the run with 100 new tokens. In this case, a relatively large deviation can be observed when comparing the official result from the challenge with our previous best result. In this regard, it is worth mentioning that, in addition to the inherent difficulty of the task itself, our model was not optimized for the individual F1 measure over the toxic class but for F1 weighted over the two classes: toxic and non-toxic. This result has been considered satisfactory since it has been obtained by means of the proposed vocabulary extension method.

Table 4. DETOXIS-Iberlef 2021 SUBTASK 2 Top 5 ranking

Ranking	Team Name	CEM	RPB	Pearson	Accuracy
	Gold_standard	1.000	0.8213	1.0000	1.0000
1	SINAI	0.7495	0.2612	0.4957	0.7654
2	Team Sabari	0.7428	0.2670	0.5014	0.7464
3	DCG	0.7300	0.3925	0.4544	0.7329
4	**GTH-UPM**	**0.7256**	**0.1545**	**0.4298**	**0.7318**
5	GuillemGSubies	0.7189	0.2449	0.4451	0.6835

If we move on to the second subtask, our result in Table 4 corresponds to the run with 75 new tokens, thus also demonstrating the convenience of the proposed extension method. In this case, in spite of the mismatch between the optimization/evaluation parameters (i.e. the subtask aims at CEM evaluation

metric while our model was specifically trained targeting weighted-F1) our model has achieved a significantly better result than the best result that we previously obtained.

7 Conclusions

In this paper, we have presented an exploratory analysis in which different deep learning (DL) models for the detection of toxic expressions have been evaluated on the DETOXIS-IberLEF 2021 challenge using the official release of the NewsCom-TOX corpus. Particularly, we have compared traditional RNN and state-of-the-art transformer models including standard BERT [5] and its BETO variant [4]. Our experiments have confirmed that optimum performance can be obtained from transformer models. Specifically, better performance has been achieved by simply fine tuning the BETO model for the toxicity detection tasks.

As another important contribution of this work, we have proposed and validated a simple but effective method for extending our pre-trained models with domain-specific vocabulary. The method accounts for term frequencies to rank and select specific words to be added at the word level. As a result, the performance of the extended model can be significantly improved. This approach could be particularly attractive to ad-hoc and special-purpose or very specific domains with unique vocabularies where limited training data is available. Nonetheless, additional work still needs to be done with regards to automatically finding or identifying the exact and optimal amount of new tokens to be added (i.e. the precise value that achieves a good balance between model complexity and available training data).

Furthermore, unlike traditional approaches not using DL nor embedding representations, when using transformer-like models and testing different text preprocessing methods, it has been observed that preserving the raw structure of the texts, by not removing anything while simply performing a very basic text normalisation, helps achieving optimal performance.

Funding The work leading to these results has been supported by the Spanish Ministry of Economy, Industry and Competitiveness through the CAVIAR (MINECO, TEC2017-84593-C2-1-R) and AMIC (MINECO, TIN2017-85854-C4-4-R) projects (AEI/FEDER, UE). Ricardo Kleinlein's research was supported by the Spanish Ministry of Education (FPI grant PRE2018-083225).

Acknowledgments We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for part of this research.

References

1. Detoxis-IberLEF 2021 results (2021), https://detoxisiberlef.wixsite.com/website/evaluation-results, [Online; accessed 21-June-2021]

2. Baziotis, C., Nikolaos, A., Chronopoulou, A., Kolovou, A., Paraskevopoulos, G., Ellinas, N., Narayanan, S., Potamianos, A.: Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. Proceedings of The 12th International Workshop on Semantic Evaluation (2018). https://doi.org/10.18653/v1/s18-1037, http://dx.doi.org/10.18653/v1/S18-1037
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017). https://doi.org/10.1162/tacl_a_00051, https://www.aclweb.org/anthology/Q17-1010
4. Cañete, J., Chaperon, G., Fuentes, R., Ho, J.H., Kang, H., Pérez, J.: Spanish pretrained bert model and evaluation data. In: PML4DC at ICLR 2020 (2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://www.aclweb.org/anthology/N19-1423
6. Fernández-Martínez, F., Griol, D., Callejas, Z., Luna-Jiménez, C.: An approach to intent detection and classification based on attentive recurrent neural networks. In: Proc. IberSPEECH 2021. pp. 46–50 (2021). https://doi.org/10.21437/IberSPEECH.2021-10, http://dx.doi.org/10.21437/IberSPEECH.2021-10
7. Garneau, N., Leboeuf, J., Lamontagne, L.: Predicting and interpreting embeddings for out of vocabulary words in downstream tasks. CoRR **abs/1903.00724** (2019), http://arxiv.org/abs/1903.00724
8. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (May 2018), https://www.aclweb.org/anthology/L18-1550
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735, https://doi.org/10.1162/neco.1997.9.8.1735
10. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: spaCy: Industrial-strength Natural Language Processing in Python (2020). https://doi.org/10.5281/zenodo.1212303, https://doi.org/10.5281/zenodo.1212303
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
12. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019), http://arxiv.org/abs/1907.11692
13. Maslej-Krešňáková, V., Sarnovský, M., Butka, P., Machová, K.: Comparison of deep learning models and various text pre-processing techniques for the toxic comments classification. Applied Sciences **10**(23) (2020). https://doi.org/10.3390/app10238631, https://www.mdpi.com/2076-3417/10/23/8631

14. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space (2013), http://arxiv.org/abs/1301.3781
15. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS 2017 Workshop on Autodiff (2017), https://openreview.net/forum?id=BJJsrmfCZ
16. Pavlopoulos, J., Malakasiotis, P., Androutsopoulos, I.: Deep learning for user comment moderation. In: Proceedings of the First Workshop on Abusive Language Online. pp. 25–35. Association for Computational Linguistics, Vancouver, BC, Canada (Aug 2017). https://doi.org/10.18653/v1/W17-3004, https://www.aclweb.org/anthology/W17-3004
17. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X.: Pre-trained models for natural language processing: A survey. Science in China E: Technological Sciences **63**(10), 1872–1897 (Oct 2020). https://doi.org/10.1007/s11431-020-1647-3
18. Rajapakse, T.C.: Simple transformers. https://github.com/ThilinaRajapakse/simpletransformers (2019)
19. Schmidt, A., Wiegand, M.: A survey on hate speech detection using natural language processing. In: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. pp. 1–10. Association for Computational Linguistics, Valencia, Spain (Apr 2017). https://doi.org/10.18653/v1/W17-1101, https://www.aclweb.org/anthology/W17-1101
20. Tai, W., Kung, H.T., Dong, X., Comiter, M., Kuo, C.F.: exBERT: Extending pre-trained models with domain-specific vocabulary under constrained training resources. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 1433–1439. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.129, https://www.aclweb.org/anthology/2020.findings-emnlp.129
21. Taulé, M., Ariza, A., Nofre, M., Amigó, E., Rosso, P.: Overview of the detoxis task at iberlef-2021: Detection of toxicity in comments in spanish. Procesamiento del Lenguaje Natural **67** (2021)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf