Human Language Technologies Project Report

Michele Morisco (505252)
Department of Computer Science, University of Pisa, Pisa, Italy
Human Language Technologies, Academic Year 2021/2022

***Abstract*—The following document aims to illustrate the work done for the Human Language Technologies course project, in which I have implemented and compared different models to perform sub-task A of the SemEval-2019 Task 6 on Identifying and Categorizing Offensive Language in Social Media [1].**

I. Introduction

Over the years, we have seen the growth of anti-social online behaviors, including cyberbullying, trolling, and offensive language. Automatic offensive language detection using machine learning algorithms becomes one solution to identifying such hostility and has shown promising performance.
The SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media dataset collects tweets through Twitter API and annotates them hierarchically regarding offensive language, offense type, and offense target.
This task is divided into three sub-tasks but I have focused on the first of them: a) detecting if a post is an offensive (OFF) or not (NOT). The macro-F1 metric independently evaluates the sub-tasks, i.e. $\frac{1}{N}\sum_{i=1}^{N} F1Score_i$ where i is the class index and N is the number of classes.
The challenges of this task include: a) the small dataset makes it hard to train complex models; b) the characteristics of language on social media pose difficulties such as out-of-vocabulary words and ungrammatical sentences; c) the distribution of target classes is imbalanced and inconsistent between training and test data.
The report is divided into three parts: in Section 2, I describe the dataset and the details of preprocessing, and the methodology of models used; in Section 3, I show the experimental results I achieved. Lastly, in Section 4, I give a brief conclusion that describes my experience.

II. Data and Methodology

A. Data description

Offensive Language Identification Dataset (OLID) is a large collection of English tweets annotated using a hierarchical three-layer annotation model. It contains 14,100 annotated tweets divided into a training partition of 13,240 and a testing partition of 860.
The distribution of the labels in OLID is shown in Table I in sub-task A.

Label	Train	Test	
OFF	4,400	240	
NOT	8,840	620	
Total	13,240	860	14,100

TABLE I
Distribution of label combinations in OLID

B. Preprocessing

Given the data that could use emojis, hashtags, and other stuff, I used some approaches to normalize the tweets to feed my models.

1) Emoji translation: First, I substituted the emojis with their corresponding text representation using a GitHub emoji project[1] that could map the emojis' unicode to the substituted phrase. In this way, we have such terms into common English words thus they could maintain their semantic meanings.

2) Tokenize tweets: Secondly, I tokenized the tweets using Ekphrasis' SocialTokenizer[2]. This tokenizer normalized some terms like email, numbers, and dates. [2] Especially, it performed word segmentation on hashtags in order to detect whether the hashtag contains profanity words. For example, #WhiteLivesMatter becomes 'White Lives Matter' which is offensive in this context.

3) Other stuff: Lastly, I converted all the text into lowercase. I removed the 'URL' tag with 'http' since 'URL' does not have embedding representation in some pre-trained embedding and models. In addition, I noticed that in several tweets there are more consecutive '@USER' tags, I limited them to one time to reduce the redundancy.

C. Methodology

1) biLSTM: Bidirectional Long Short-Term Memory (biLSTM) is a recurrent neural network used primarily on natural language processing, the input flows in both directions, and it's capable of utilizing information from both sides. It's also a powerful tool for modeling the sequential dependencies between words and phrases in both directions of the sequence. I would also like to use LSTM as my powerful baseline model to compare and report the result.
To evaluate the performance of the model, I tested different configurations of the network and I used two different word embeddings obtained by two different pre-trained word vectors (GloVe): glove with 6B tokens on Wikipedia corpus and glove with twitter data and 27B tokens. The specific setting is shown in Table II.

2) BERT: Google's research team releases Bidirectional Encoder Representation from Transformer (BERT) [3] and achieves state-of-the-art results on many NLP tasks. BERT is

[1]https://github.com/carpedm20/emoji
[2]https://github.com/cbaziotis/ekphrasis

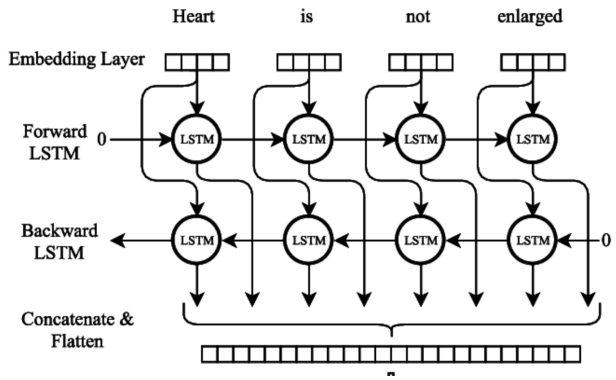

Fig. 1. Architecture of biLSTM

GloVe Type	Tokens	Vocabulars size	Dimension
Twitter.27B.100d	27B	1.2M	100
6B.100d	6B	400K	100

TABLE II
TYPE OF GLOVE USED FOR BILSTM MODELS

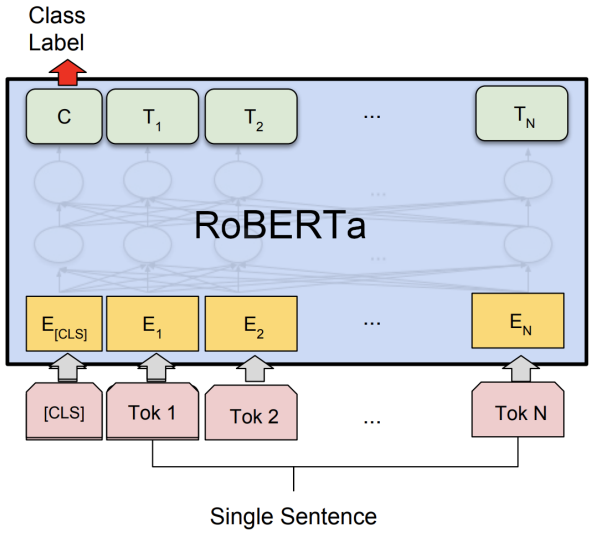

Fig. 2. Architecture of RoBERTa

a transformers model pre-trained on a large corpus of English data, I tested the different sizes of BERT models [4] [5], and the various structures are shown in Table III.

BERT	Layer	Hidden	Heads
Tiny	2	128	2
Small	4	512	8
Medium	8	512	8
Base	12	768	12

TABLE III
STRUCTURES OF DIFFERENT BERT VERSIONS

3) BERTweet: BERTweet [6] is trained based on the RoBERTa pre-training procedure. The corpus used to pre-train BERTweet consists of 850M English tweets. This model has the same architecture as $\text{BERT}_{\text{Base}}$ which is trained with a masked language modeling objective. According to the authors, BERTweet performs better than $\text{RoBERTa}_{\text{Base}}$.

4) DistilBERT: DistilBERT [7] is a small, fast, cheap, and light Transformer model based on the BERT architecture. Knowledge distillation is performed during the pre-training phase to reduce the size of a BERT model by 40%. According to the authors, DistilBERT is 60% faster than BERT but it retains 97% of BERT performance. As such, DistilBERT is distilled on very large batches leveraging gradient accumulation using dynamic masking and without the next sentence prediction objective.
In general, it has the same general architecture as BERT, but the token-type embeddings and the pooler are removed while the number of layers is reduced by a factor of 2. In my experiment, I tested its capability compared with other transformers.

5) RoBERTa: Robustly Optimized BERT Pretraining Approach (RoBERTa) [8] builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates. RoBERTa has the same architecture as $\text{BERT}_{\text{Large}}$ (L = 24, H = 1024, A = 16, 355M parameters) but uses a larger byte-level BPE (Byte-Pair-Encoding) as a tokenizer and uses a different pretraining scheme. BPE is a hybrid between character- and word-level representations that allow handling the large vocabularies common in natural language corpora. Instead of full words, BPE relies on subwords units, which are extracted by performing statistical analysis of the training corpus.
In a few words, RoBERTa is trained with:

- dynamic masking
- FULL-SENTENCES without NSP loss
- large mini-batches
- larger byte-level BPE

I tested RoBERTa for checking its performance between different capabilities, in detail, I tested three versions of RoBERTa:

- $\text{RoBERTa}_{\text{Base}}$
- $\text{RoBERTa}_{\text{Base}}$ trained on $\sim 58M$ tweets.
- $\text{RoBERTa}_{\text{Base}}$ trained on $\sim 58M$ tweets and finetuned for offensive language identification with the TweetEval benchmark

III. RESULTS

In my experiments, I tested different configurations of the models described in the previous section. In this part, describe the experimental results obtained from my models with each of their validation accuracies.

A. *biLSTM*

In both versions I used the same following hyperparameters:

- Number of units: 64
- Number of hidden layers: 3
- Dropout: 0.5
- Batch size: 128

Tweet	True	Prediction
@USER @USER Put DeLauro in a police lineup identifying the bag lady" - she would be picked everytime! She has to be proof either Conn voters are incompetent to vote or she is part of a vast voter fraud conspiracy! No one votes for a woman that gross!"	OFF	NOT
#RAP is a form of ART! Used to express yourself freely. It does not gv the green light or excuse the behavior of acting like an animal! She is not in the streets of the BX where violence is a way of living. Elevate yourself boo and get on @USER level for longevity! #QUEEN	NOT	OFF
#StopKavanaugh he is liar like the rest of the #GOP URL	OFF	NOT

TABLE IV
EXAMPLES OF WRONG PREDICTION ON ROBERTA$_{\text{OFFENSIVE}}$

Both use the same optimizer NAdam, the Adam algorithm with Nesterov momentum, but with different learning rates and epochs. BiLSTM-27B used a learning rate equal to 0.001 and 4 epochs; on the contrary, BiLSTM-6B used 0.0001 and 3 epochs.
These hyperparameters are the results of the grid-search. The following Table V summarizes the hyperparameters used with the best validation results.

Model	Macro-F1	Accuracy
biLSTM-27B	0.7444	0.7723
biLSTM-6B	0.6838	0.7255

TABLE V
RESULTS ON DEVELOPMENT SET IN THE BILSTM MODELS

These results show that the first model with Twitter word vectors seems to perform better with respect to the standard one. The results of the Twitter model could be improved thanks to more words and tokens, especially using some social-based words. Indeed, the standard word vectors use fewer words with respect to the other one.
Despite the results, I was going to make a more depth grid-search but the small amount of data and the small computing capacity of Colab's notebook did not allow me to do because that is expensive in terms of time and space.

B. BERT, DistilBERT and RoBERTa

During the experiments, I tested different configurations to improve the validation accuracy which is shown in Table VI.

Model	Learning rate	# Epochs	Batch
BERT$_{\text{Tiny}}$	1e-3	3	32
BERT$_{\text{Small}}$	2e-4	3	32
BERT$_{\text{Medium}}$	2e-5	3	32
BERT$_{\text{Base}}$	2e-6	5	32
BERTweet	2e-6	7	4
DistilBERT	2e-6	5	32
RoBERTa$_{\text{Base}}$	2e-6	4	8
RoBERTa$_{\text{Twitter}}$	2e-6	4	8
RoBERTa$_{\text{Offensive}}$	2e-6	3	8

TABLE VI
HYPERPARAMETERS USED IN THE VARIOUS TRANSFORMERS MODELS

I applied a K-fold Cross-Validation with k equal to 4. I had to set a less batch size in RoBERTa models compared to other ones because using a high size raised an error on CUDA memory. The same situation with BERTweet which used more resources, indeed, I had to reduce the batch size to 4 and use less maximum length sequence, of 32 rather than 64.
They were trained using the same BERT$_{\text{Large}}$ architecture, which probably that uses more resources of CUDA to run them.

Model	Macro-F1	Accuracy
BERT$_{\text{Tiny}}$	0.7130	0.7628
BERT$_{\text{Small}}$	0.7340	0.7709
BERT$_{\text{Medium}}$	0.7506	0.7801
BERT$_{\text{Base}}$	0.7538	0.7858
BERTweet	0.7614	0.7907
DistilBERT	0.7483	0.7789
RoBERTa$_{\text{Base}}$	0.7604	0.7863
RoBERTa$_{\text{Twitter}}$	*0.7674*	*0.7915*
RoBERTa$_{\text{Offensive}}$	**0.7922**	**0.8148**

TABLE VII
RESULTS ON DEVELOPMENT SET IN THE VARIOUS TRANSFORMERS MODELS

In Table VII, we can see the results achieved on the development set and we notice the smaller size BERT versions get less accuracy with respect to the Base version, as expected. In addition, DistilBERT has less accuracy with respect to BERT$_{\text{Medium}}$ in this task; on the contrary, we can see that RoBERTa$_{\text{Base}}$ gains a good score with respect to BERT$_{\text{Base}}$. Indeed, for this task, RoBERTa seems the best model reaching an accuracy equal to 0.786.
BERTweet, as mentioned in the authors' statement, is slightly better than RoBERTa$_{\text{Base}}$, which becomes the best model for this task.
But the last two models, as described in the previous section, are pre-trained on the TweetEval benchmark and one is finetuned for offensive language identification. These models are specialized versions of Roberta and we immediately notice their advantages compared to using more generic transformers, as we are going to see, they yielded better results.

C. Final results

By analyzing the validation results, is possible to conclude that the BERT models performed better than biLSTM, according to the results of other papers about this task, I was expecting this behavior.
Given these results, the winner is RoBERTa$_{\text{Offensive}}$, as expected but I decided to check also the 2nd winner, i.e. RoBERTa$_{\text{Twitter}}$.

In Table VIII, we can see the results of the test set.

Model	Macro-F1	Accuracy	OFF	NOT
$\text{RoBERTa}_{\text{Twitter}}$	0.8025	0.8465	0.7093	0.8957
$\text{RoBERTa}_{\text{Offensive}}$	0.8099	0.8500	0.7226	0.8972

TABLE VIII
RESULTS ON TEST SET WITH BEST MODELS

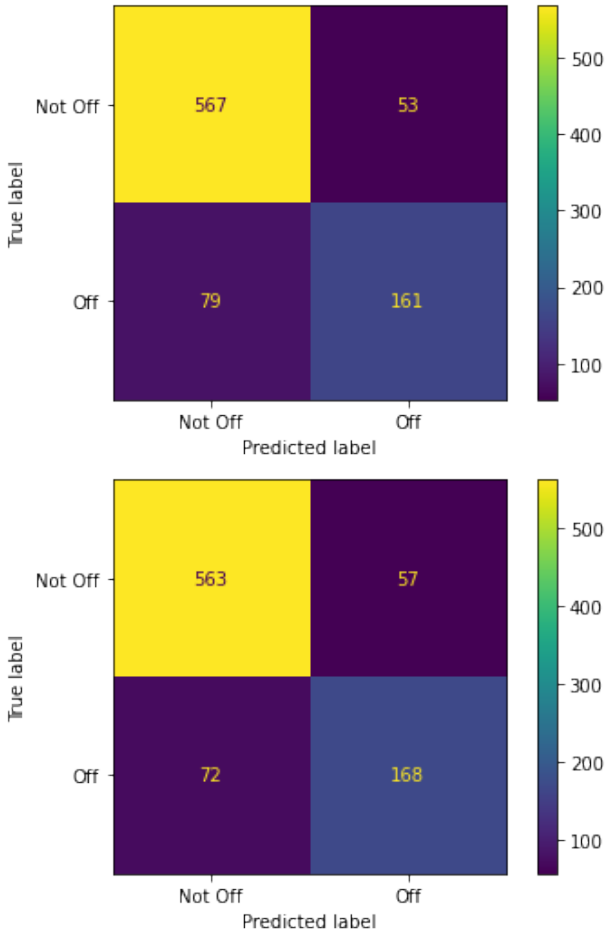

Fig. 3. $\text{RoBERTa}_{\text{Twitter}}$ (above) and $\text{RoBERTa}_{\text{Offensive}}$ (below) after fine-tuning

We can see that offensive tweets are less recognized by models compared to not offensive ones. This behavior is natural because the dataset is strongly unbalanced, indeed together with the limited data present on the dataset, this is a reason why the other models performed badly compared to the winning ones. In detail, we can see Fig.3 which confirms the score about the offensive and not offensive tweets' accuracies. Table IV is shown some examples of tweets predicted incorrectly and I understand how important it is to include the context in a sentence. In addition, the tweets are often written with incorrect grammar or use some internet slang that makes it difficult to recognize the meaning.
To do these experiments, I used a Tesla T4 GPU on Colab and a Tesla P100-PCIE-16GB on Kaggle.

IV. CONCLUSION

I developed a system for classifying a sentence as an offensive speech, which usually depends on the ideology and the social context. In addition, in a social network is complicated to recognize the context and meaning then it is more important to learn how to include in the semantics the context of it.
During my experiments, I tested different methods comparing old NLP models such as Bi-LSTM with new ones like pre-trained transformers. In the end, I achieved good results with RoBERTa versions and BERTweet models classifying in the top ten positions in the SemEval-2019 Task 6: OffensEval.
Naturally, the $\text{RoBERTa}_{\text{Offensive}}$ model is classified in the fourth position, a very good score, but for a specialized model is expected this result. The project made me realize that this research field can be challenging, but it is still evolving and improving its capability.

REFERENCES

[1] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval)," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, 2019, pp. 75–86.

[2] C. Baziotis, N. Pelekis, and C. Doulkeridis, "Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, August 2017, pp. 747–754.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[4] P. Bhargava, A. Drozd, and A. Rogers, "Generalization in nli: Ways (not) to go beyond simple heuristics," 2021.

[5] I. Turc, M. Chang, K. Lee, and K. Toutanova, "Well-read students learn better: The impact of student initialization on knowledge distillation," *CoRR*, vol. abs/1908.08962, 2019. [Online]. Available: http://arxiv.org/abs/1908.08962

[6] D. Q. Nguyen, T. Vu, and A. T. Nguyen, "BERTweet: A pre-trained language model for English Tweets," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 9–14.

[7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692