HUMAN LANGUAGE TECHNOLOGIES

Francesco Corti
f.corti3@studenti.unipi.it

Giovanni Sorice
g.sorice@studenti.unipi.it

Department of Computer Science, University of Pisa, Pisa, Italy

Human language technology

Academic Year: 2020/21

March 29, 2021

ABSTRACT

The following document aims to illustrate the work done for the Human Language Technologies course project. We implemented and compared three neural network models to perform the EVALITA HaSpeeDe2 Hate Speech Detection task. At the end of the experiments, we noticed how it is important to consider the context of a sentence. Furthermore, we achieved the 76% on the test set. In the end, we understood how challenging and evolving the Natural Language Processing research field is.

1. Introduction

The Internet has allowed people across the world to communicate instantaneously, this has had a positive influence on society. However, it has also broadened the potential for harm; hateful messages are amplified on the social media in ways that were not previously possible. The purpose of this project is to build an Hate Speech Detector that given a tweet is able to recognize, if it contains hateful content or not. The dataset used to build the model is obtained from Evalita-2020, a challenge to evaluate the NLP and Speech tools for the Italian language, more in details on the haspeede2 task. In order to use the tools and the cloud computing capacity offered by Google, the models were developed and deployed on Google Colab. This report is structured as follows: in Section 2, the dataset and the preprocessing done on it are described. In Section 3, the architecture of the models used is shown and commented. In Section 4, the performance of the models are compared and commented. In Section 5 an experiment on Twitter data retrieved by a scraper is commented. In the end (Section 6) a brief conclusion that describes our overall experience is written.

2. Dataset

The dataset was given by the EVALITA HaSpeeDe 2020 task organizers. The dataset includes texts targeting minority groups such as Immigrants, Muslims and Roma communities, whose relative social problems constantly feed the public and political debate triggering hate speech. In this perspective, tweets and newspapers headlines using specific keywords related to the mentioned Italian minorities were collected. The development dataset (training + validation) is composed of 5723 rows and the test set is composed of 443

news and 1263 tweets.
The importance of a good dataset is well known to everyone, however having a good dataset with a catastrophic preprocessing phase will lead models to bad results. For this reason, we took advantages of previous years tasks and researches. From the "Preprocessing" section of [1] we obtained an initial baseline for our preprocessing phase. It has the following structure:

- Extraction of the first feature: length of the comment;
- Extraction of the second feature: percentage of words written in CAPS-LOCK inside the comment;
- Removing Tags;
- Replace the characters ‘&’, ‘@’ respectively in the letters ‘e’, ‘a’;
- Conversion of disguised bad words;
- Extraction of the third feature: Hashtag used;
- Hashtag splitting;
- Removing URL;
- Extraction of the fourth feature: number of ‘?’ or ‘!’ inside the comment;
- Extraction of the fifth feature: number of ‘.’ or ‘,’ inside the comment;
- Punctuation removal;
- Removal of nearby equal vowels, removal of nearby equal consonants if they are more than 2;
- Translation of emoticons;
- Replacement of the abbreviations with the respective words;
- Removal of the laughs;
- Extraction of the sixth feature: number of bad words in the comment;
- Extraction of the seventh feature: percentage of bad words.

3. Models

Since 2012, the Natural Language Processing research community, has started to use machine learning models based on neural networks, to tackle language processing problems. Following this trend we analyzed three machine learning models, the first one is the bidirectional LSTM, the second one is the Kim Convolutional neural network and the last one based on BERT model is AlBERTo. To avoid to learn the correlation between sentences position and their output, we performed a shuffle of the dataset at the beginning of the training phase.

3.1 biLSTM

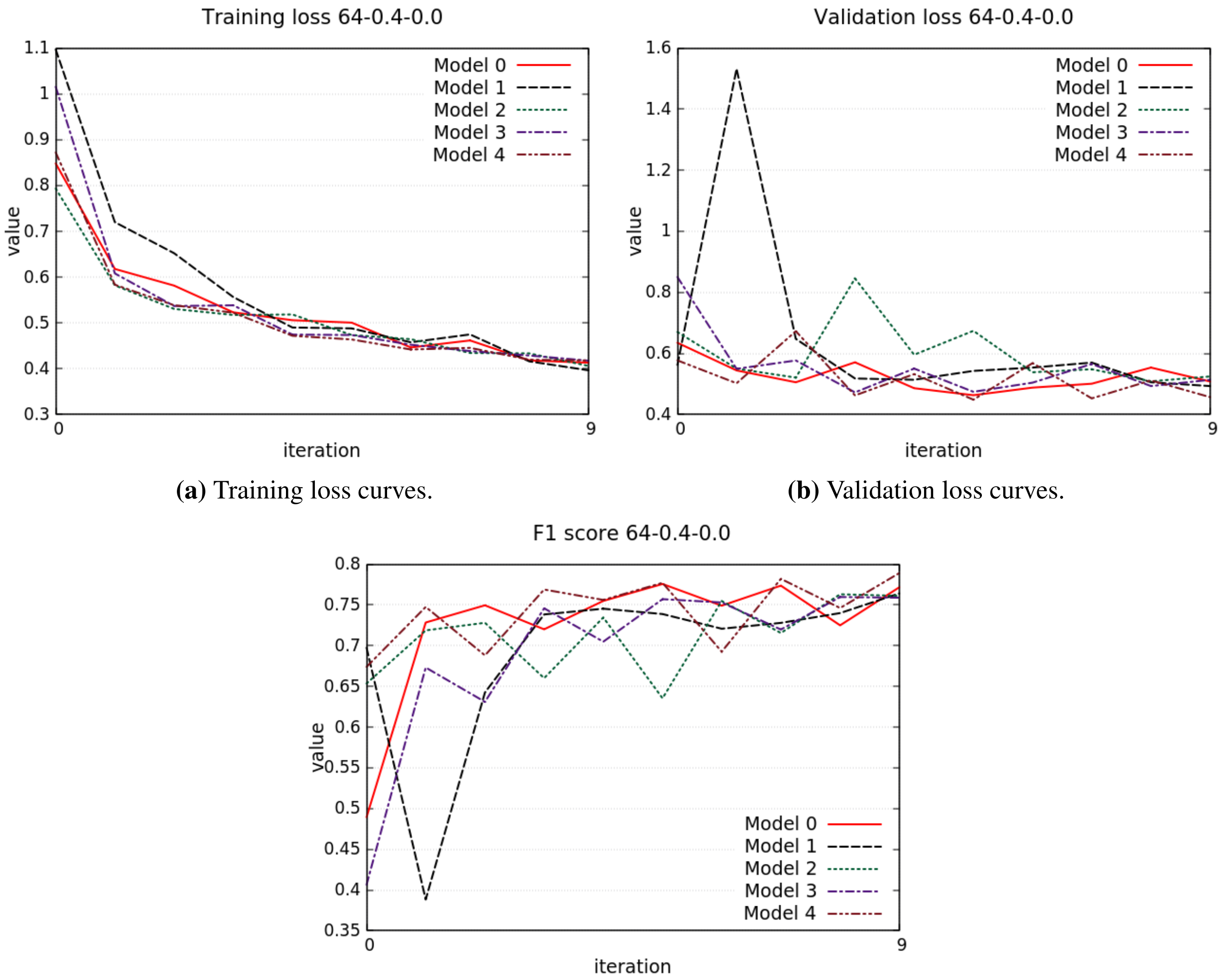
Before the introduction of the Bert model, the bidirectional LSTM based models, were among the most used model to tackle NLP problems. Now their popularity is decreased but they continue to be a good baseline for performance comparisons between models. Two type of model were built and tested, models with a pre-trained embedding layer and models without it. The results showed a better performance in the models with the pre-trained embedding layer and for this reason we took in consideration only these models. Different configuration of the network were tested, with one or more bidirectional LSTM layer and with different configuration for the dense part. The choice of the final model was done by taking in consideration that a bigger model doesn't guarantee the best solution. The final models are composed by two bidirectional LSTM layer and three dense layer of 256, 64 and 32 units. Dropout and L2 regularization were used to avoid overfitting and to reduce the complexity of the models. A first grid search was done on the following parameters:

- number of units in the bidirectional LSTM layer 64, 128, 256, 512;
- dropout in the bidirectional LSTM layer 0, 0.2, 0.4, 0.6, 0.8;
- L2 regularization on the dense layer 0.0, 0.0002, 0.0004, 0.0006, 0.0008.

From this grid search, the first four selected models parameter were used to train four new networks with the k-fold cross validation with a k set to 5. The results are shown in the table 3.2.

#Units bilstm	Dropout	L2 regularization	Error TR	Error VS	F1 VS
64	0.2	0.0	0.29422	0.57164	0.76111
64	0.4	0.0	0.41024	0.49942	0.76892
128	0.6	0.0	0.46864	0.52894	0.73995
256	0.6	0.0002	0.47004	0.51182	0.76027

Table 3.1: Best network configurations with binary cross-entropy error and F1 score.



(a) Training loss curves.

(b) Validation loss curves.

(c) F1 score curves.

Figure 3.1: K-fold Training curves of Bidirection LSTM models with 64 units, 0.4 of dropout and 0.0 of L2 regularization.

3.2 CNN

CNN are mostly applied to analyze visual imagery. Also, CNN can be a good model to tackle structured data, like time series and sequence in general. For this reason, we decided to try a specific model of CNN meant for NLP problems: KimCNN [2]. The main idea behind the KimCNN is to model combinations of 2,3, ... , n words. Accordingly to this, a sentence can be visualized like an image and not like a concatenation of words, this can be view in fig 3.2.

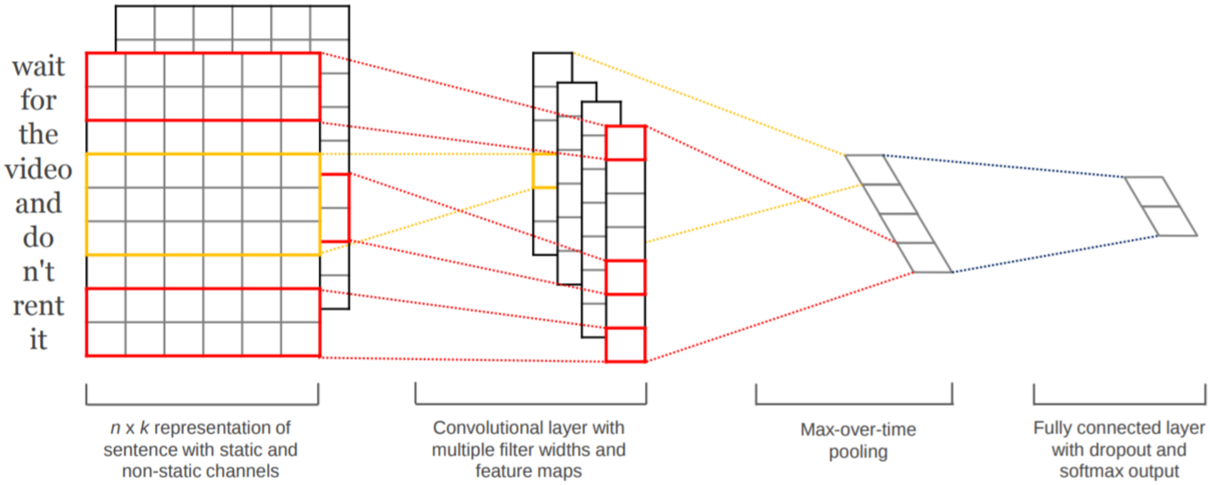

wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 3.2: The architecture of KimCNN.

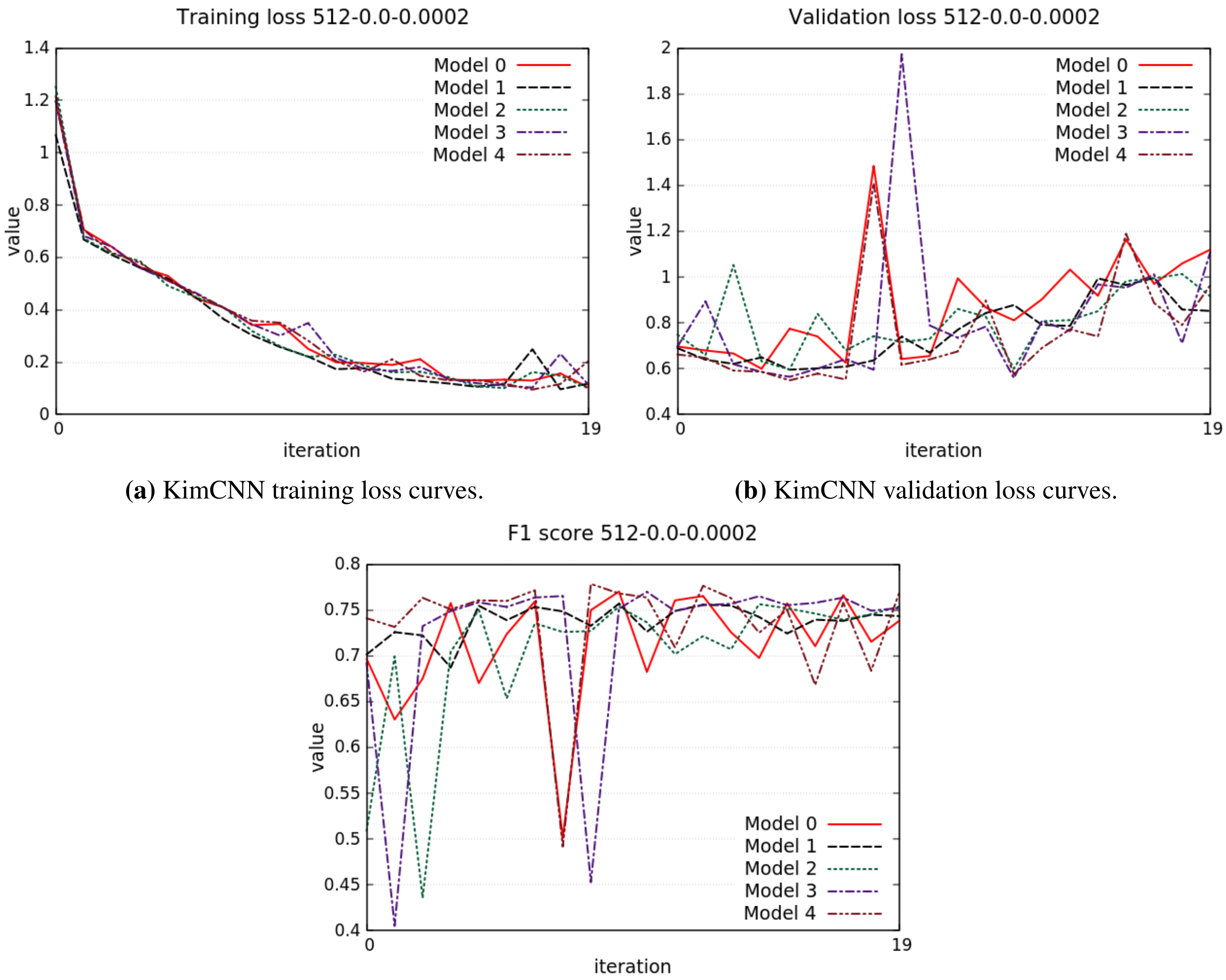
Multiple configurations of the network were tested, with one or more different size for the convolution filter, and with different configuration for the dense part. The choice of the final model was done by taking in consideration the validation values and the risk of overfitting for each models. The final models are composed by three 2D convolution layer with filters size of 3, 4 and 5. Next, there is a max pooling layer, a dropout layer and at the end, there are three dense layer of 256, 128 and 32 units. Dropout and L2 regularization were used to avoid overfitting and to reduce the complexity of the models. The optimizer used is Nadam and the loss is the binary crossentropy. Before the grid search for the final hyper-parameter, some trials were done to understand the best configuration of number for the convolutions layers, the size of the filters and the number of epochs. From these trials, we decided to use three 2D convolution layer with filter size of 3,4 and 5 trained for 20 epochs. A grid search was done on the following parameters:

- number of filter for each convolution layer 64, 128, 256, 512, 1024;
- L2 regularization on the convolution layer 0.0, 0.0002, 0.0004, 0.0006, 0.0008;
- L2 regularization on the dense layer 0.0, 0.0002, 0.0004, 0.0006, 0.0008.

From the grid search, the first four selected models parameter were used to train four new network with the k-fold cross validation with a k set to 5. The results are shown in the table 3.2.

#Filter	L2 CNN	L2 dense	Error TR	Error VS	F1 VS
128	0.0002	0.0	0.20858	0.96784	0.73262
512	0.0	0.0	0.04744	0.89992	0.75121
512	0.0	0.0002	0.12806	0.99186	0.75122
1024	0.0	0.0	0.05852	0.88502	0.73738

Table 3.2: Best network configurations with binary cross-entropy error and F1 score.



(a) KimCNN training loss curves.

(b) KimCNN validation loss curves.

(c) KimCNN F1 score curves.

Figure 3.3: K-fold Training curves of KimCNN models with 512 filters, 0.0 of L2 regularization for the convolution part and 0.0002 of L2 regularization for the dense part.

3.3 AlBERTo

In the past years, the NLP community research has proven the outstanding effectiveness of models such as ELMo, GPT and BERT. In particular AlBERTo [3], the model that we used, is a pretrained BERT [4] language understanding model for the Italian Language, it is trained by using a large number of Italian tweets retrieved from TWITA (a corpus of Tweets in the Italian language collected from February 2012 to nowadays from Twitter's official streaming API). AlBERTo is composed of twelve stacked Transformer Encoder [5] layers, each of them has an encoder architecture using stacked self-attention and point-wise, fully connected layers for the encoder. BERT models are trained using masked learning and the next following sentence techniques. In the former the fifteen percent of each sequence terms is masked at random, then the neural network is trained to predict those masked terms. In the latter, the model is trained to learn the relationship between sentences by predicting the next sentence of a previous sentence given as input.

However, AlBERTo was originally trained using only the masked learning technique because it is used with Tweets data, and we do not have cognition of a flow of tweets as it happens in a dialogue that has questions and answers. Because of this, AlBERTo is not suitable for question answering tasks but it is suited for classification and prediction tasks. The architecture of AlBERTo can be visualized in the figure 3.4.

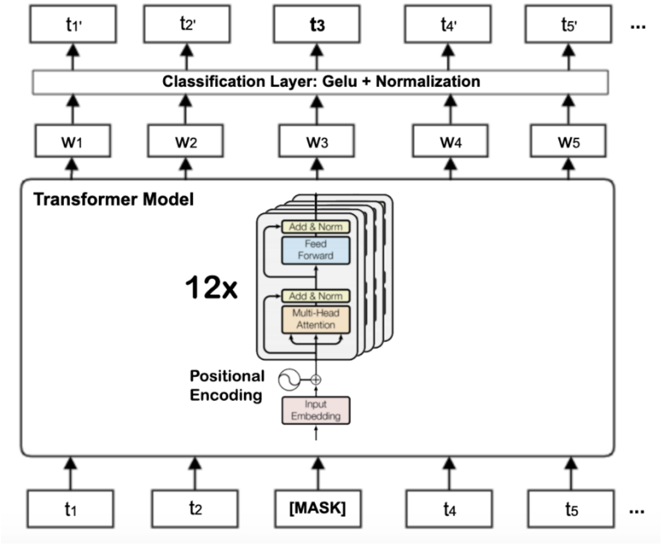
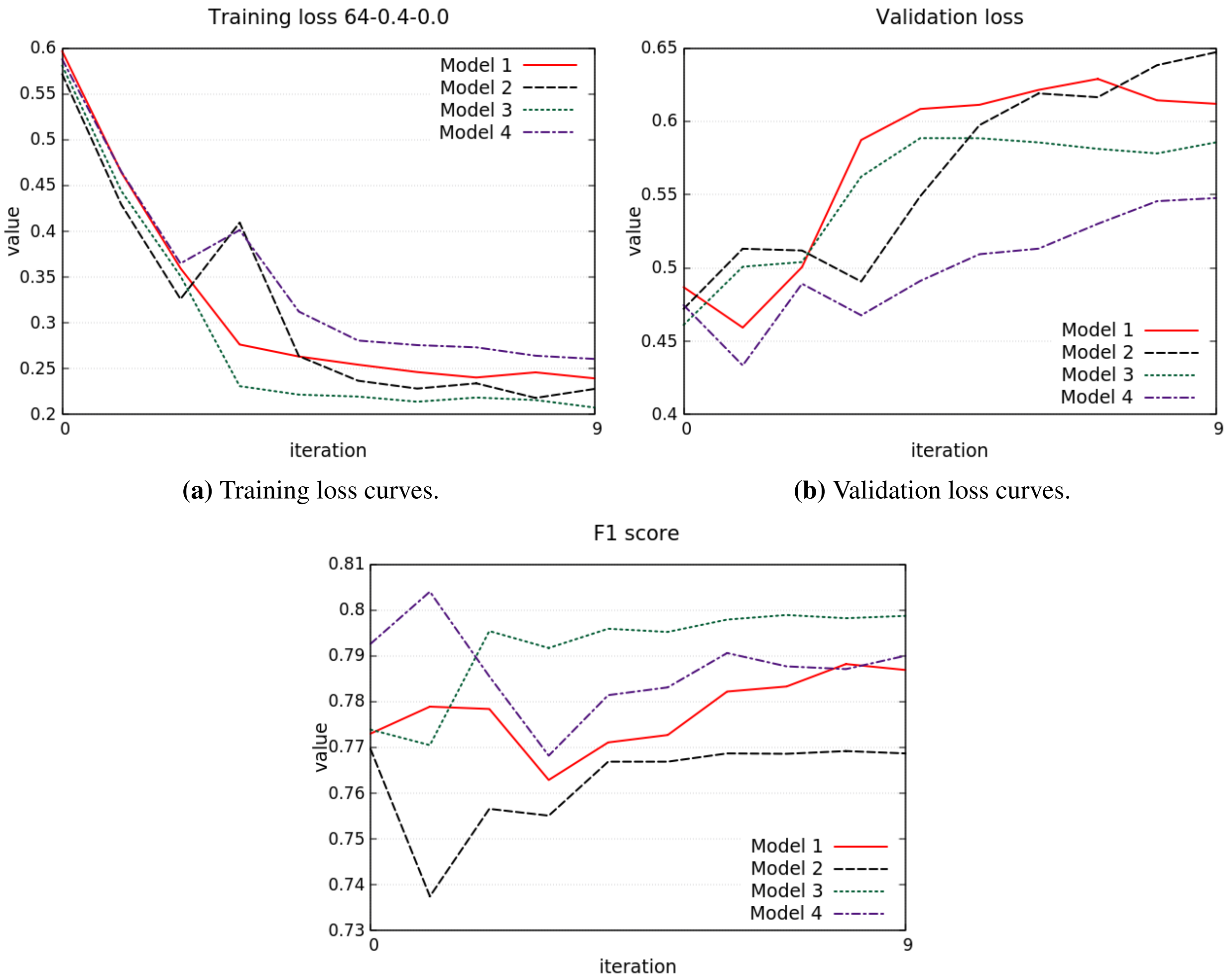


Figure 3.4: AlBERTo model architecture.

BERT models implementation consist of two phases, pre-training and fine-tuning. In the former, the model is trained on unlabeled data over different pre-training tasks. In the latter, the model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks. To achieve a good performance on the hate detection task, we performed a fine-tuning on the AlBERTo model. To understand how many epochs of training the model needs to perform in a good way the hate speech detection task. We tried different way of fine-tuning. Starting from the full training of the model to the training of the final classification dense part only. We realized that ALBERTo needs first to be fine-tuned on the full model and then to be trained on the last dense part. So, three train epochs were done on the whole model, these were followed by seven train epochs in which only the last dense connected part was trained, while the rest of the model parameters were kept constant. Since AlBERTo model was trained with data preprocessed in a specific way, we processed our datasets with the same preprocessing algorithm. Otherwise, the fine-tuning procedure would have invalidated some of the features already extracted with the original train. AlBERTo preprocessing differs from our preprocessing, it does not extract artificial feature (e.g. extracting the size of the sentence, extracting the number of bad words contained) but it leaves this task to the BERT model. Also, during BERT preprocessing they do not convert the disguised and wrong written words. The results obtained are shown in the plots below fig.3.5.



(a) Training loss curves.

(b) Validation loss curves.

(c) F1 score curves.

Figure 3.5: K-fold curves of AlBERTo.

4. Results

Model	Error TR	Error VS	F1 VS	F1 TS tweets	F1 TS news
bi-LSTM	0.41024	0.57164	0.76111	0.76032	0.68316
KimCNN	0.12806	0.99186	0.75122	0.74178	0.66399
Alberto	0.26066	0.54763	0.79	0.7615	0.69882
Baseline_MFC	N\A	N\A	N\A	0.3366	0.3894
Baseline_SVC	N\A	N\A	N\A	0.7212	0.621

Table 4.1: Best network configurations with binary cross-entropy error and F1 score on validation and test sets.

By analysing the table 4.1 is possible to conclude that Bert model performed slightly better than the others. After reading the results of others paper about this topic, we were expecting this behavior, but also we were expecting a bigger performance gap between Bert model and the other two models. In our opinion, this is due to two main reasons: first the training set has a small amount of data and second because of the small computing capacity of Colab's notebooks. These two things combined, did not allow us to do a bigger hyperparameters grid search for a model that is expensive in term of time and space. Moreover, we found in [6] that the winner of the Haspeede2 competition, added to the original training dataset a "silver" dataset of 20 thousands hate speech labeled tweets. We reasoned about that choice and found out that model like ALBERTo needs a huge amount of data to perform at its best. As expected the results of the BiLSTM and KimCNN were better than the model baselines. Surprisingly, they also are in the range of other submitted BERTs model's results, this strengthen our hypothesis that BERTs model need a huge amount of data and computational power to perform at their best.

5. Hate speech application

In order to understand the usability the models we developed in a real-world scenario, we decided to test them on some Twitter's account. By using a scraper for Twitter [7], we created a simple pipeline for the evaluation of the hate speech levels of some Twitter's users. For privacy reasons, we decided to omit the users' name from the tables 5.1 and 5.2.

Tweet	KimCNN	BiLSTM	ALBERTo
cavolo ma #pedullà sta in tutte le trasmissioni Mediaset... abbiate pietà, levatelo di torno	1	1	0
ho scritto giorni fa che gli insulti contro la #Meloni erano solo l'inizio di una campagna di discredito contro una che non ha niente da nascondere. E infatti... Merde, diffamatori, delinquenti... non ci spaventate, alla fine sarete cancellati	0	0	1
Se ne sentiva la mancanza, adesso sì che abbiamo risolto tutti i problemi dell'Italia. https://tgcom24.mediaset.it/2021/video/roma-murale-con-bacio-tra-due-donne-e-il-primo-in-italia_29323673-02k.shtml…	0	1	0
Vengo come penitente che chiede perdono al Cielo e ai fratelli per tante distruzioni e crudeltà; come pellegrino di pace, in nome di Cristo, Principe della Pace. Quanto abbiamo pregato, in questi anni, per la pace in #Iraq! Dio ascolta sempre. Sta a noi camminare nelle sue vie.	1	1	1
#AstraZeneca, ora avanti tutta col piano vaccinale, seguendo i modelli Figliuolo e Bertolaso. Serve fare in fretta. Non sono tollerabili altri errori da parte di Bruxelles: aspettiamo il licenziamento di tutti quelli che hanno sbagliato.	1	1	0
No a sconti per mafiosi all'ergastolo. Galera e nessun regalo per chi non si pente e non collabora, come suggerì Falcone. Lo Stato non può arrendersi, vanificando anni di impegno contro organizzazioni criminali. Lo dobbiamo alle vittime, a chi combatte i mafiosi, ai nostri figli.	1	1	1

Table 5.1: Tweets with the label produced by the models.

User	KimCNN	BiLSTM	ALBERTo	avg
Politician account	15/224	21/224	5/224	14/224
Local political party account	8/51	7/51	7/51	8/51
Common account 1	6/48	4/48	4/48	4/48
Common account 2	10/130	6/130	6/130	7/130
Religious exponent account	7/64	23/64	10/64	13/64

Table 5.2: Tweet accounts with total number of hate speech tweets detected.

After analyzing the results, we understood how important it is to include the context in a classified sentence. Moreover, the social and political context heavily influence the semantic of a sentence. For this reason, the models need to be constantly updated to follow these changes. Furthermore, all the information has to be exploited for the classification task. As you can see in the third entry of the table 5.1, understanding a link content can be crucial to classify a sentence accurately.

6. Conclusion

Classify a sentence as hate speech is a subjective task that usually divides the society depending on the ideology and the social context. Based on this, hate speech detection is really difficult to accomplish, because it is strongly conditioned from the biases inserted in the training dataset. As observed in §5 it is extremely important to take into consideration the social and political context of a sentence. For this reason, it is not useful to consider only the syntax and the semantics of a text but it is more important to learn how to include in the semantics the context of it. Taking into consideration that we did not use an external dataset, we would have been classified in the top ten positions in the Evalita HaSpeeDe2 challenge. In the end, the project gave us the opportunity to compare old NLP models with new cutting edge NLP models on a subjective task. The project made us realize that this research field is still evolving and improving to achieve human mind capability.

References

[1] Lorenzo Ferri Giulio Bianchini and Tommaso Giorni. Text analysis for hate speech detection in italian messages on twitter and facebook.

[2] Yoon Kim. Convolutional neural networks for sentence classification.

[3] Marco de Gemmis Giovanni Semeraro Valerio Basile Marco Polignano, Pierpaolo Basile. Alberto: Italian bert language understanding modelfor nlp challenging tasks based on tweets.

[4] Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. Bert: Pre-training of deep bidirectional transformers forlanguage understanding.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[6] Elisa di Nuovo Simona Frenda Marco Stranisci Cristina Bosco Tommaso Caselli Viviana Patti Irene Russo Manuela Sanguinetti, Gloria Comandini. Haspeede 2 @ evalita2020: Overview of the evalita 2020 hatespeech detection task.

[7] Yassine Ait Jeddi. Scweet.