SSL - Milestone 3

Ionescu Diana & Dumitrescu Andrei

1. Introduction

For the third milestone of the SSL project we have experimented with different datasets, built different baselines and tried out a state-of-the-art approach. Our initial roadmap for the project was made out of 4 steps: deciding on which datasets to use in our experiments, build different baselines, experiment with 2 different state-of-the-art models researched and finally, combine the main ideas from the state-of-the-art models to build a new architecture. For this milestone we have experimented with 3 different datasets, built 2 different baselines and 1 of the 2 state-of-the-art models. In the following paragraphs we will showcase our experiments and the obtained results. Also, our preliminary implementation and experiments can be found at the following link: https://github.com/AndreiDumitrescu99/SSL_Project.

2. Datasets

As mentioned in the previous milestone we mainly decided to experiment with different models on how they perform on different social media contexts such as: comments, tweets or conversations and also found three potential datasets to represent these contexts. The datasets found are: Conv Abuse (made out of conversations with chatbots), Toxic Comment Datasets (made out of tweets) and Abusive Language on YouTube (made out of YouTube comments). During this milestone we have analysed these datasets and experimented with them.

The Abusive Language on YouTube (*ALYT* for short*)* dataset version that we used is made out of samples split into 3 different categories: non-abusive, abusive and uncertain / neutral. It contains a total of 19915 samples, out of which 17122 samples (85.97%) are considered not-abusive, 2274 (11.41%) abusive and 519 (2.60%) uncertain / neutral. The dataset is not properly balanced having a clear bias to the non-abusive samples. Also, the dataset doesn't come with an official split between train / valid / test sets. We decided to do this by ourselves. In order to achieve somewhat "balanced" subsets, we firstly split it into the three categories represented by the labels. Using this split we proportionally divide them into the training set and testing set. The proportion used is 20% as it is mentioned in the paper. This percentage of samples is taken from all 3 subsets to form the testing set. After this process the training set contains 15931 samples and the testing one 3981 samples. Similarly, we shuffle the train set and split it once more into a final training part and a validation part. Also, the validation part is made out of 20% of the samples from the original train set.

The ConvAbuse dataset comes already with the dataset split. It is a two-labelled dataset, the labels being: abusive and non-abusive. The training set consists out of 2501 samples out of which 338 (13.5%) are considered abusive, the validation set is made out of 831 samples from which 112 are abusive (13.47%) and lastly, the test set consists of 853 samples with 128 abusive ones (15%). As it can be seen, this dataset is also unbalanced and quite small, but the percentage of abusive samples is constant through all the splits. A sample usually is made out of 4 messages between a human and a chatbot and multiple labelings from different annotators. The dataset offers an overall label for each sample and this is the

final label that we will use in our experiments. Also, to represent a sample we concatenate all the 4 messages in a single string.

Lastly, the Toxic Comment Datasets (*TCC* for short) is actually a collection of different datasets that include annotated tweets. Most of these datasets are in English, but some of them are in different languages. We decided to discard the sets which don't use English. Also, being a collection of multiple sets, there are multiple labels. We decided that all the samples that are labelled with "none" will be considered non-abusive and the rest of them as abusive. The dataset is not already split into three parts, so we apply the same procedure as before, resulting in 54821 samples for training and 13705 samples for testing. From all the samples, around 76% are considered offensive in this case.

3. Baselines

In the case of baselines, we decided to represent the data using word2vec word embeddings. For each text sample we built an average "sentence" embedding which is actually a vector made out of the average of the word vectors of the words composing the samples. Each word embedding has a size of 300, so the final vector representation is also of size 300 per sample.

As baselines we decided to use a Support Vector Machine Classifier and a classic Neural Network. The SVM used is the one from the Scikit Learn library and uses the default parameters for it. To implement the Neural Network we used the TensorFlow API. We decided to use these APIs as we are already familiar to them. Another alternative for the Neural Network implementation would be the PyTorch API. The Neural Network proposed is made out of 5 Dense Layers, which gradually have fewer units. The first 4 layers have as activation function the ReLU function and the last one has the Softmax function. This architecture is trained for 100 epochs, with a batch size of 64, using the Adam Optimizer with a learning rate of 5e-5. The last Dense Layer has a total of 2 or 3 units depending on the dataset. The loss function that we used is the Categorical Cross-Entropy. We experimented with different hyper-parameters but these were the best we have found. We run experiments with each baseline for each dataset. The Neural Network was trained using the training and validation splits and only the models that performed the best regarding the validation loss were kept. In all of these cases, our architecture converged. To make sure of this we have plotted graphs as shown in the following two figures. Figure 1 presents the evolution of the loss values of the training set and validation set of the *ALYT* dataset during the training process. Figure 2 presents similarly the evolution of the accuracy score on the same dataset. To measure the performance of our models we used the *Macro F1-Score* and the *Accuracy Score.* The results will be presented in the ***Results*** section in the following paragraphs.

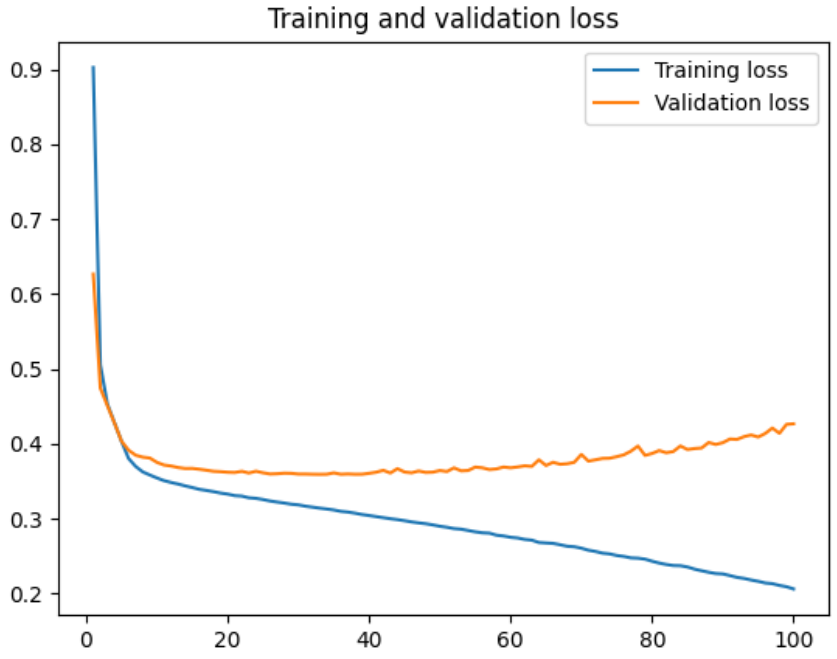

Figure 1 - Loss Evolution of the Neural Network Baseline on ALYT dataset

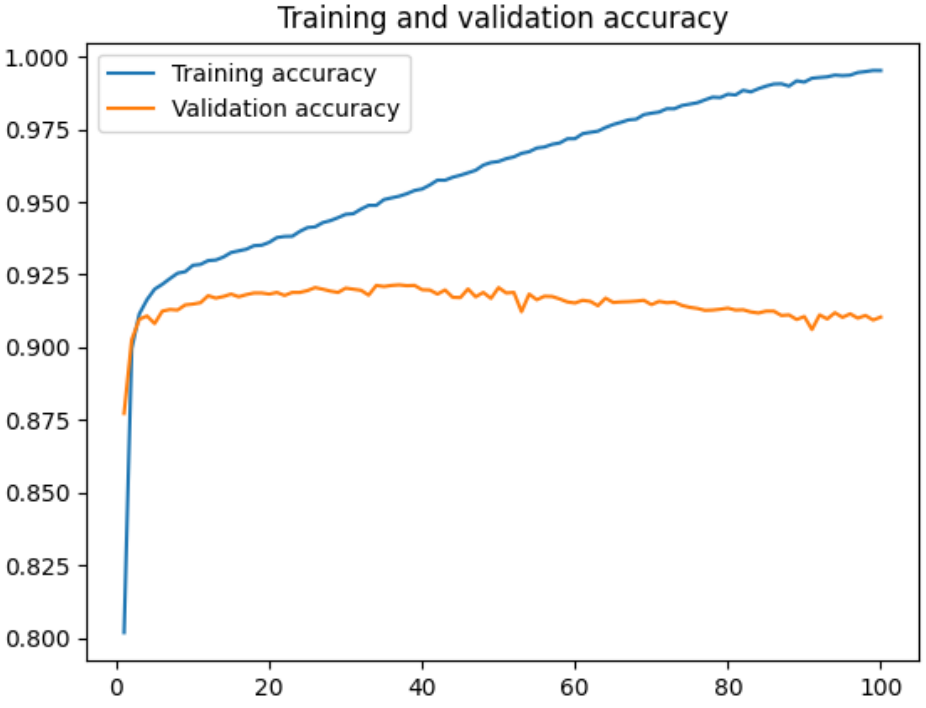

Figure 2 - Accuracy Evolution of the Neural Network Baseline on ALYT dataset

4. Bi-directional LSTMs with Self-Attention

For this milestone we also wanted to experiment with one state-of-the-art architecture presented in the last milestone. We decided to start with the idea from the *Pay "Attention" to*

Your Context when Classifying Abusive Language Research Paper. It proposes an architecture based on Bi-directional LSTMs and on a type of attention called *self-attention.* In the case of these experiments we represented our samples as a list of word vectors, where each word vector corresponds to a word. By doing this, a sample will have the shape of *(Number Of Words, Embedding Size).* We decided to use at maximum 120 words per sample in each dataset. The samples which don't have so many words are padded with vectors full of 0s and samples which have more than 120 words are dropped.

There is a public implementation of this architecture, but the custom attention seems to be implemented with a custom layer and *Theano* backend. To simplify this we decided to reimplement using TensorFlow API the attention mechanism by taking into consideration the original implementation. In our case, the attention is computed using a Dense Layer with a Tanh activation function and TensorFlow functions.

We decided to keep almost all the architecture of layers as proposed in the original implementation. The architecture is made out of: 2 bi-directional LSTM Layers, 3 Dropout Layers, 2 Dense Layers and the Attention mechanism part. Similarly to the baseline Neural Network, the last Dense Layer has Softmax activation function. The architecture uses Adam Optimizer with a learning rate of 1e-4, the loss function is Categorical Cross-Entropy and the training process lasts for 50 epochs with a batch size of 128. The convergence of the models was verified using the same procedure as above with plotting graphs. Results will be presented in the following section.

5. Results

In this section we would like to present our intermediary results. As mentioned above, the metrics used are Accuracy and Macro F1-Score. Macro F1-Score can be a better metric because the class distribution on our datasets are unbalanced. Table 1 showcases our results on each model and on each dataset. In the table, the best results obtained by our models are bolded. The bi-directional LSTM model with self-attention performs best on almost all metrics on all datasets, except on the Macro F1-score on the TCC dataset.

Table 1 - Results of our Experiments

	SVM Baseline		***Neural Network Baseline***		***Bi-directional LSTMs with self attention***	
	Macro F1	*Accuracy*	*Macro F1*	*Accuracy*	*Macro F1*	*Accuracy*
ALYT	0.41	0.87	0.47	0.88	**0.48**	**0.88**
ConvAbuse	0.67	0.88	0.78	0.89	**0.82**	**0.90**
TCC	**0.63**	0.71	**0.63**	0.70	0.60	**0.77**

6. Further Work

The next step in our project is to make experiments with another SOTA of this task, namely HateBERT. It means that we will want to fine tune this model for each dataset we

used previously. If the results are the desired ones, we will want to create a new architecture combining the HateBERT model with the self-attention concept introduced by the previous SOTA presented in this milestone.

Moreover, we will want to have a second look over the datasets, because we think that some samples can be mislabeled. Also, we will compute the confusion matrix to analyse the qualitative results of our models. Lastly, we would like to document our implementation as for now it doesn't have much documentation.

7. References

[1] Curry, A. C., Abercrombie, G., & Rieser, V. (2021). ConvAbuse: Data, Analysis, and Benchmarks for Nuanced Abuse Detection in Conversational AI. *arXiv preprint arXiv:2109.09483*.

[2] Bertaglia, T., Grigoriu, A., Dumontier, M., & van Dijck, G. (2021, August). Abusive Language on Social Media Through the Legal Looking Glass. In Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021) (pp. 191-200).

[3] Risch, J., Schmidt, P., & Krestel, R. (2021, August). Data integration for toxic comment classification: Making more than 40 datasets easily accessible in one unified format. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)* (pp. 157-163).

[4] Chakrabarty, T., Gupta, K., & Muresan, S. (2019, August). Pay "attention" to your context when classifying abusive language. In *Proceedings of the Third Workshop on Abusive Language Online* (pp. 70-79).

[5] Caselli, T., Basile, V., Mitrović, J., & Granitzer, M. (2020). Hatebert: Retraining bert for abusive language detection in english. arXiv preprint arXiv:2010.12472.