Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200602

Pretraining and Fine-Tuning Strategies for Sentiment Analysis of Latvian Tweets

Gaurish THAKKAR [a,1] and Mārcis PINNIS [b,c]
[a] *Faculty of Humanities and Social Sciences, University of Zagreb,*
Ul. Ivana Lučića 3, 10000, Zagreb, Croatia
[b] *Tilde, Vienības gatve 75A, Riga, Latvia, LV-1004*
[c] *University of Latvia, Raiņa bulv. 19-125, Riga, Latvia, LV-1586*

Abstract. In this paper, we present various pre-training strategies that aid in improving the accuracy of the sentiment classification task. At first, we pre-train language representation models using these strategies and then fine-tune them on the downstream task. Experimental results on a time-balanced tweet evaluation set show the improvement over the previous technique. We achieve 76% accuracy for sentiment analysis on Latvian tweets, which is a substantial improvement over previous work.

Keywords. Sentiment analysis, word embeddings, BERT, Latvian

1. Introduction

Sentence-level sentiment analysis (SA) aims to classify the opinion expressed by the author into either a Positive, Negative, or Neutral class. Recently, transformer-based neural networks [1] pre-trained using self-supervision [2, 3] have shown state-of-the-art performance on downstream tasks [4]. However, most of these findings have been reported for highly resourced languages.

In this work, we focus on improving the performance of SA for Latvian tweets using a model of pre-trained multilingual Bidirectional Encoder Representations from Transformers (mBERT). We experiment further by pre-training the model with in-domain data. mBERT treats Unicode emoticons as out-of-vocabulary words. We propose adding them to the vocabulary of the model and repeating the pre-training and fine-tuning cycle. We also compare A-Lite-BERT (ALBERT) [5] and ELECTRA [6] models as light-weight variants of mBERT which we train from scratch on Latvian tweets. We release all the pre-trained language representation models and the models along with the code[2].

2. Related Work

Pre-trained word-embeddings [7, 8] have been studied extensively for improving sentiment classification scores [9]. For Latvian, Pinnis [10] performed experiments on Lat-

[1] Corresponding Author: Gaurish Thakkar; E-mail: gthakkar@m.ffzg.hr.
[2] https://github.com/thak123/bert-twitter-sentiment

vian tweets with a wide range of classifiers and features. Peisenieks and Skadiņš [11] analysed machine translation as a viable tool for performing SA for Latvian tweets. A recent study [12] performed pre-training of the BERT model from scratch for classifying sentiment of tweet representations. Earlier techniques [13, 14] used Pointwise Mutual Information (PMI) with Information Retrieval (IR) as well as Naive Bayes to classify multi-domain tweets.

3. Pre-training and Fine-tuning Strategies

We follow 3 different pre-training strategies:

- First, an existing pre-trained model trained on a large corpus is trained further on the in-domain corpus.
- Second, the model trained using the previous method is trained further by adding new tokens into the existing vocabulary. Our initial experiments showed that emoticons are treated as unknowns (*[UNK]*) as they are not present in the vocabulary of the pre-trained model. Since the mBERT model was trained on texts from Wikipedia, it is obvious to lack smileys in the text. We hypothesise that emoticons are sentiment-bearing tokens and hence important as features.
- Lastly, we pre-train (ALBERT and ELECTRA) models from scratch. The models may have vocabularies learned from the data or they may use vocabularies from existing models. We perform this step to compare the performance of pre-trained models with the models that are trained from scratch.

Using the various annotated datasets, we perform fine-tuning on the downstream task of sentiment analysis using all the models described previously.

4. Data

In our experiments, we use the sentiment annotated corpora curated by Pinnis [10]. The corpora are:

1. *Gold*: a corpus consisting of 6,777 human-annotated Latvian tweets from the period of August 2016 till November 2016.
2. *Peisenieks*: a corpus consisting of 1,178 human-annotated Latvian tweets created by Peisenieks and Skadiņš [11].
3. *Auto*: three sets of tweets from the period of August 2016 till July 2018 automatically annotated based on sentiment-identifying emoticons that are present in the tweets – 23,685 tweets with emoticons, 23,685 tweets with removed emoticons, and 47,370 tweets with both present and removed emoticons.
4. *English*: a corpus of 45,530 various human-annotated English tweets from various sources that were machine-translated into Latvian.
5. A time-balanced evaluation set that consists of 1,000 tweets from the period of August 2016 till July 2018.

To pre-train word embeddings, we use also the Latvian tweets from the Latvian Tweet Corpus[3] [10]. The corpus consists of 4,640,804 unique Latvian tweets that have been collected during the time-frame from August 2016 till March 2020.

5. Experiments

In this section, we describe the experimental setup for sentiment analysis. Our experimental setup consists of pre-training and fine-tuning steps. We perform the following pre-processing steps on the text:

1. Tokenization.
2. Removal of URLs.
3. Replacement of consecutive user mentions with a single mention.
4. User mention replacement with a placeholder ('*mention_i*' where the *i* stands for the i^{th} mention in the tweet).
5. Lower-casing of the whole tweet.

5.1. Pre-training

We employ the script[4] available in the *transformers*[5] library to continue training the uncased version of the multilingual-BERT (mBERT) model. mBERT models 102 languages, which also include Latvian and other Baltic languages. This step uses the 4.6 million unique tweets from the Latvian Tweet Corpus described above. The corpus is split into train and eval and is pre-trained for 7 epochs. In the case of unknown tokens, there are around 5 thousand unique *[UNK]* tokens in the Gold dataset (train split only), which mainly are emoticons. Therefore, we sort the highest occurring emoticons and add 70 of them to the model vocabulary. Then, we perform one more cycle of pre-training with the new vocabulary in the network. Using the same Latvian Tweet Corpus, we train two more models namely ALBERT and ELECTRA from scratch[6]. This is done by tokenizing the whole corpus and joining each of the two consecutive tweets together as examples to be trained. These are used to train a discriminator to decide if each token in the corrupted input was replaced by a generator sample or not. Both embedding_size and hidden_size are set to 256.

All models use the same vocabulary as that of the pre-trained uncased mBERT model, which uses sentence-piece [15] as the method of tokenization and word splitting. We use a batch size of 16. For the pre-training step, the process was stopped once the perplexity score of ≈ 3 was achieved on the validation split of the dataset.

5.2. Fine-tuning

For this step, We have the following pre-trained language representation models:

[3] https://github.com/pmarcis/latvian-tweet-corpus

[4] https://github.com/huggingface/transformers/blob/master/examples/language-modeling/run_language_modeling.py

[5] https://github.com/huggingface/transformers

[6] https://github.com/shoarora/lmtuners/tree/master/lmtuners

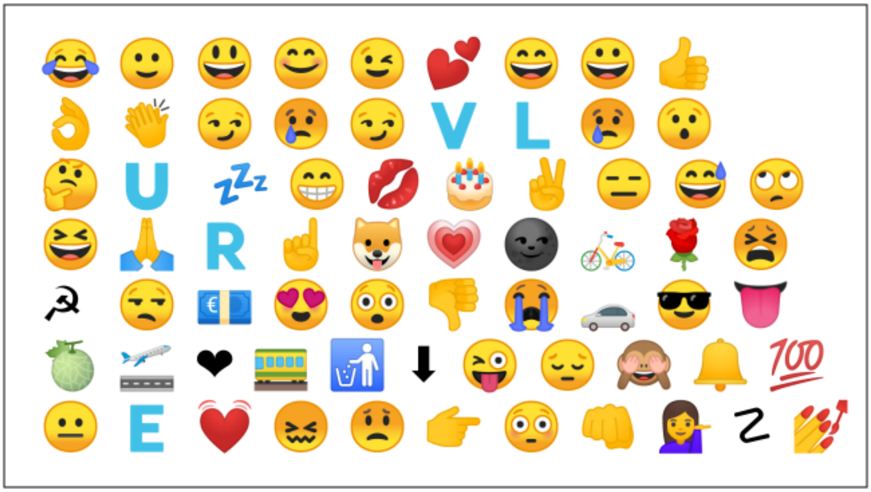

Figure 1. Examples of (non-exhaustive) list of added emoticons

1. mBERT - vanilla version.
2. mBERT - pre-trained on the Latvian Tweet Corpus.
3. mBERT - pre-trained on the Latvian Tweet Corpus plus emoticons added to the vocab.
4. ALBERT and ELECTRA.

For each tweet, the vector representing the [CLS] token is extracted and passed to the classification layer. All settings use a single 3-class softmax classification layer (*positive, negative, and neutral*) with a dropout value of 0.2. We employ a maximum sequence length of 150 and pad the shorter tweets. We randomly sample 1000 records as the validation set. Finally, we report the test accuracy on the model with the highest validation accuracy. No hyper-parameter tuning on the model is performed. Except for the emoticon augmented model, all other representation models share the same vocabulary.

6. Results

The accuracy results of the experiments are presented in Table 1. We compare our scores with previously reported scores by [10] (shown in the first column). The column named 'Base' shows the results of using the mBERT model directly in the fine-tuning task. The next column 'Pre' lists scores when the model is additionally pre-trained with in-domain Twitter data. The results show that there is an improvement over the Perceptron baseline when the mBERT model is used. There is also further improvement when we pre-train the existing model with the in-domain corpus. The results of the experiment where we incorporate emoticons into the vocabulary are presented in column 'Pre+Emo'. This setup improves over the in-domain pre-training setup except in two cases.

The best results were obtained when the *Gold* and *Auto (with ☺)* datasets were combined to train the sentiment analysis model. We see a drop in performance for all models trained on the *Gold+Auto (no ☺)* and *Gold+Auto (both)* datasets. Even though

Table 1. Results of the classifier (Accuracy Scores)

Dataset	Perceptron [10]	mBERT			ALBERT	ELECTRA
		Base	Pre	Pre+Emo		
Gold	0.661	0.678	**0.756**	0.754	0.661	0.711
Gold+Peisenieks	0.676	0.692	0.747	**0.764**	0.698	0.706
Gold+Auto (with ☺)	0.624	0.679	**0.769**	0.748	0.649	0.68
Gold+Auto (no ☺)	0.512	0.523	0.648	**0.660**	0.483	0.621
Gold+Auto (both)	0.487	0.526	0.618	**0.657**	0.509	0.564
Gold+English	0.613	0.698	0.692	**0.720**	0.669	0.684

ELECTRA has a lower number of model parameters (compared to mBERT), it is still able to perform better than the vanilla mBERT version.

7. Error Analysis

We performed error analysis using the best-performing sentiment analysis model (i.e., the mBERT model that was additionally pre-trained on the Latvian Tweet Corpus and fine-tuned using the *Gold+Auto(with ☺)* corpus. To aid the error analysis, we visualised the test set by plotting the individual tweet representations and their predictions as scatter plots. For every tweet in the test set, we use the *[CLS]* token, which is a vector of length 768, and project it down to 50 dimensions using Principal Component Analysis (PCA) [16]. The principal components are further reduced to 2 dimensions using t-SNE [17]. Each of the points is plotted as nodes. We color each of the correctly predicted tweets in green for positive, red for negative, and blue for neutral. The incorrect predictions are colored in black with the correct class and predicted class as the node text. The visualisation is depicted in Figure 2.

We started the error analysis by investigating whether we can identify if messages that are grouped in clusters that are formed in the tweet representation and prediction scatter plot have common characteristics (e.g., common syntactic structures). This did not yield positive results as the messages close to each other often contained different syntactic structures and even different vocabularies. Therefore, we continued by analysing what common characteristics can be found among a random subset of 100 misclassified tweets. From the analysis, we made the following observations:

- 32 of the misclassified tweets were ambiguous to the extent where external world knowledge would be necessary to decide on the polarity of the messages.
- 17 of the misclassified tweets featured words of the wrongly selected polarity within the messages, which may indicate that the model may have learned to use lexical polarity-identifying cues to aid classification. However, it would require further analysis to validate this hypothesis.
- 13 of the misclassified tweets featured sarcastic expressions within the messages. All of the sarcastic tweets were negative tweets. This amounts to almost 50% of all misclassified negative tweets.
- 12 of the misclassified tweets featured possible multiple polarities within the messages.
- 4 tweets featured double negation.

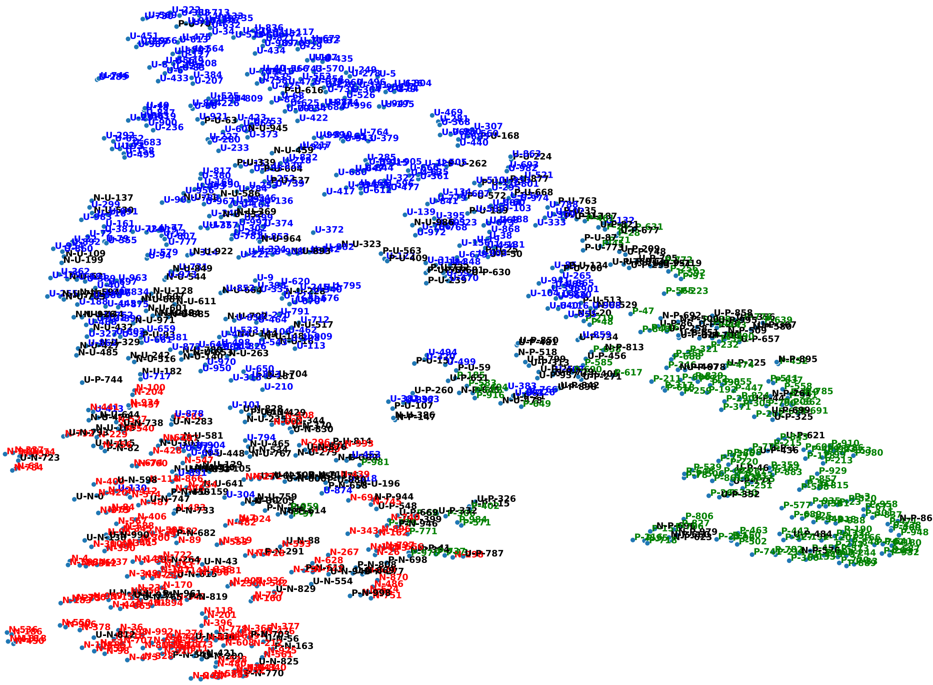

Figure 2. Tweet representation and prediction scatter plot

- 3 tweets featured spelling mistakes and a lack of diacritics that could have triggered misclassification.
- For the remaining 19 tweets, we did not identify common characteristics.

8. Conclusion

In this paper, we presented our work on improving Latvian SA for tweets. Our experiments allowed us to achieve the increase in performance when pre-training word embedding models with in-domain unlabelled data and fine-tuning the models on relatively small supervised datasets. The results surpass previous work on SA for Latvian. As future work, handling tweets with mixed emotions will be investigated. Furthermore, error analysis indicated that a large proportion of misclassified tweets can be attributed to ambiguous and sarcastic tweets for which analysis and consideration of tweet history could potentially allow expanding the context available for classification and, thereby, allow performing better-informed classification. Error analysis also raised a hypothesis that the fine-tuned models may have learned to focus on lexical polarity-identifying cues when deciding on which class to assign to tweets. This needs to be validated in further research. Lastly, there are still avenues of improvements to ELECTRA model pre-training evident that have not been explored and could be investigated in future work.

9. Acknowledgments

The work presented in this paper has received funding from the European Union's Horizon 2020 research and innovation programme under the *Marie Sklodowska-Curie* grant *agreement no. 812997* and under the name CLEOPATRA (Cross-lingual Event-centric Open Analytics Research Academy). This research has been supported by the European Regional Development Fund within the joint project of SIA TILDE and University of Latvia "Multilingual Artificial Intelligence Based Human Computer Interaction" No. 1.1.1.1/18/A/148. The work has been carried out during Gaurish Thakkar's internship in Tilde.

References

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998–6008.

[2] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.

[3] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:190711692. 2019.

[4] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:191010683. 2019.

[5] Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:190911942. 2019.

[6] Clark K, Luong MT, Le QV, Manning CD. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:200310555. 2020.

[7] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781. 2013.

[8] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 1532–1543.

[9] Severyn A, Moschitti A. Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2015. p. 959–962.

[10] Pinnis M. Latvian Tweet Corpus and Investigation of Sentiment Analysis for Latvian. In: Human Language Technologies – The Baltic Perspective - Proceedings of the Seventh International Conference Baltic HLT 2018. Tartu, Estonia: IOS Press; 2018. p. 112–119.

[11] Peisenieks J, Skadins R. Uses of Machine Translation in the Sentiment Analysis of Tweets. In: Baltic HLT; 2014. p. 126–131.

[12] Azzouza N, Akli-Astouati K, Ibrahim R. TwitterBERT: Framework for Twitter Sentiment Analysis Based on Pre-trained Language Model Representations. In: International Conference of Reliable Information and Communication Technology. Springer; 2019. p. 428–437.

[13] Gulbinskis I. Digitālo tekstu sentimenta analīze. 2010.

[14] Špats G, Birzniece I. Opinion Mining in Latvian Text Using Semantic Polarity Analysis and Machine Learning Approach. Complex Systems Informatics and Modeling Quarterly. 2016;(7):51–59.

[15] Kudo T, Richardson J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2018. p. 66–71.

[16] Wold S, Esbensen K, Geladi P. Principal component analysis. Chemometrics and intelligent laboratory systems. 1987;2(1-3):37–52.

[17] Maaten Lvd, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008;9(Nov):2579–2605.