Transformer-based Sarcasm Detection in English and Slovene Language

Matic Rašl
University of Maribor,
Faculty of Electrical
Engineering and Computer Science,
Koroška cesta 46, 2000 Maribor, Slovenia
matic.rasl@student.um.si

Mitja Žalik
University of Maribor,
Faculty of Electrical
Engineering and Computer Science,
Koroška cesta 46, 2000 Maribor, Slovenia
mitja.zalik1@student.um.si

Vid Keršič
University of Maribor,
Faculty of Electrical
Engineering and Computer Science,
Koroška cesta 46, 2000 Maribor, Slovenia
vid.kersic@um.si

Abstract

Sarcasm detection is an important problem in the field of natural language processing. In this paper, we compare performances of the three neural networks for sarcasm detection on English and Slovene datasets. Each network is based on a different transformer model: RoBERTa, DistilBert, and DistilBert – multilingual. In addition to the existing Twitter-based English dataset, we also created the Slovene dataset using the same approach. An F1 score of 0.72 and 0.88 was achieved in the English and Slovene dataset, respectively.

Keywords natural language processing, sarcasm detection, transformers, RoBERTa, DistilBert

1 Introduction

Language is the essential tool for communication in real life and online in the digital world. With the fast growth of the internet in the last two decades, an enormous amount of text data is available to everyone, which is one of the main reasons natural language processing (NLP) has become one of the fastest-growing fields in computer science and artificial intelligence. While the most commonly used NLP application is text translation, many other applications are being researched and applied, e.g., text summarization, emotion recognition, sarcasm, and irony detection [1]. In this paper, we focus on the sarcasm detection problem.

Sarcasm detection is defined as a binary classification problem, where the goal is to detect if the given text is sarcastic [2]. The most common places to find sarcastic comments are social media platforms, e.g., Twitter, where people often express their opinions and views on different topics. While in some examples, e.g., *"I work 40 hours a week for us to be this poor"*, it is easy to spot, sometimes, e.g., *"Great, that's just what I needed!"* is harder to perceive at first sight. Detection of sarcasm is essential because not understanding and detecting it can lead to substantial miscommunication errors and disagreements. Automatic sarcasm detection is also crucial in other NLP problems, such as sentiment analysis, where undetected sarcasm can negatively affect an analysis. Therefore, there is a need for automatic detection of sarcastic comments and text.

This paper compares performances of three neural networks for sarcasm detection on English and Slovenene datasets. Each neural network is based on a transformer model. In the following sections, we overview the related work, describe the used datasets, present the experiment, analyze the results, and conclude the paper emphasizing future work.

2 Related Work

Automatic sarcasm detection dates back to 2006 [3], but it has gotten momentum in the past few years with advancements in the fields of neural networks and NLP. In general, sarcasm can be detected in three different ways [2]. Rule-based approaches use specific evidence, such as words or phrases, for identification. Such techniques were often used in earlier systems, such as [4]. Statistical approaches either use text features or learning algorithms to find sarcasm. Statistical methods were used in works, such as [5], where combinations of positive verbs and negative situation phrases were used as classification features. The most common approach today is by using deep learning techniques. For example, in [6], the model can learn user-specific context and thus achieve better results than previous state-of-the-art models.

Significant advancements in NLP tasks were achieved with transformers. They are a new form of neural network that does not use convolution and recursion. Instead, they use attention to find correlations between words in the text. Transformers can process text in parallel, allowing much faster learning than sequential methods [7]. They also achieve better results than previous methods.

DOI https://doi.org/10.18690/978-961-286-516-0.10
ISBN 978-961-286-516-0

With the increasing number of learning parameters, neural networks need a larger training dataset to prevent overfitting. While building large labeled datasets can be demanding, it is easy to construct large unlabelled corpora. Therefore, large models can be trained on unlabelled text data to create a good language model, i.e., expressive word embeddings. Afterwards, these representations can be used for different NLP-related tasks [5]. The mainstream architecture of the pre-trained models is Bidirectional Encoder Representations from Transformers (BERT). The initial model was pre-trained on BooksCorpus and English Wikipedia, which advanced state-of-the-art for eleven NLP tasks [8]. Nowadays, many BERT-based architectures exist. For example, RoBERTa (A Robustly Optimized BERT Pre-training Approach) [9] optimizes the way of masking tokens and thus improving the performance of the model. Another common architecture is DistilBert [10], which has reduced the number of training parameters. That makes its training 60 % faster while retaining 97 % of BERTs language understanding capabilities.

BERT has been widely and successfully utilized for sarcasm detection [11]. In [12], the accuracy is even more improved by also considering the context of sarcastic comments. The authors in [13] use RoBERTa to detect sarcasm with even higher accuracy. Although BERT-based architectures are very successful, their pre-training still has some drawbacks. Sarcasm is present primarily in informal communication (e.g., social networks such as Reddit, Twitter, etc.), which was not part of the training set. Therefore, in [14] BERT was outperformed by the context-independent GloVe embeddings model, which was pre-trained on Twitter data.

3 Datasets

Constructing a dataset for the sarcasm detection problem is not a straightforward task since the perception of sarcasm is difficult even for people. A general approach to dataset creation is to scrap the data from different social media platforms, e.g., Twitter, Reddit, and use user-specified labels, i.e., hashtags on Twitter and */s* on Reddit [11, 15, 16]. But this approach has several drawbacks, like users not annotating sarcasm with tags or misusing labels to express their opinion better. The Headlines dataset was introduced to solve the mentioned problem. The dataset contains headlines from two news websites: one, where real-world events are reported, and the other with sarcastic descriptions of events, including sarcastic headlines [17]. The third common way is to manually label data, but this is time-consuming and still requires the annotator with a good sense of sarcasm.

Since no dataset for sarcasm detection in the Slovenian language exist, and manual labeling is time-consuming, we created the Slovene dataset with the user-specified labels. As a knowledge base for our task, tweets (i.e., posts on Twitter) were selected. Tweets, annotated by users with specific hashtags (e.g., #sarcasm, #sarkazem), were considered sarcastic (i.e., positive) examples, while other tweets were non-sarcastic (i.e., negative) examples. For the English dataset, we selected the one from the 2nd Workshop on Figurative Language Processing [11] because it was constructed in the same way as the Slovene dataset. Before training, datasets were split into the training and the test sets, as shown in Table 1.

Table 1: Train-Test split.

Set	**English**	**Slovene**
Train	5000	759
Test	1800	272

4 Method

Although transformers can be fine-tuned to specialize in a specific task, the process takes a long time on common hardware. However, as shown in [13], fine-tuning can be avoided by utilizing other networks to find correlations in transformer embeddings. Since the transformer's weights are not changing, its output can be calculated only once and then saved before learning the second part of the network. This approach significantly improves learning times.

For the experiment, we implemented the neural network model similar to the one used in [13]. As some details about the network were missing in the mentioned paper, we also relied on implementation in [18]. The architecture of the neural network is shown in Figure 1. During the experiment, three different transformers were explored, RoBERTa [9], pre-trained on English dataset, and two DistilBert [10] transformers, one pre-trained on the English language and the other one on multiple languages (DistilBert mult). DistilBert transformers are smaller than RoBERTa (66 M vs. 125 M training parameters), which translates to significant speedup embedding generation time.

As mentioned before, tokenized inputs were transformed to embeddings at the beginning of the training. Embeddings were saved to the *Transformer output cache*. Then they were used as input to the Bidirectional LSTM layer, whose outputs were concatenated with original embeddings before the pooling layer. Before flattening the data, 1D spatial dropout was applied. After dense and another dropout layer, a dense layer with softmax activation was applied.

For the training, the Google Colab [19] environment with Google TPU was used. Neural networks were trained for 25 epochs with a 10 % validation split. In the end, weights of epoch with the smallest validation loss were restored.

5 Experiments and Results

The accuracy of the models was tested on the test datasets with the embedding of length 20. Additionally, we tested the model using RoBERTa transformer with the embeddings of length 100 to find out how embeddings length affects the results. However, since the results with larger embeddings were similar to those obtained with smaller ones, we did not train DisitlBert based models on larger embeddings due to the hardware limitations. Results are shown in Table 2.

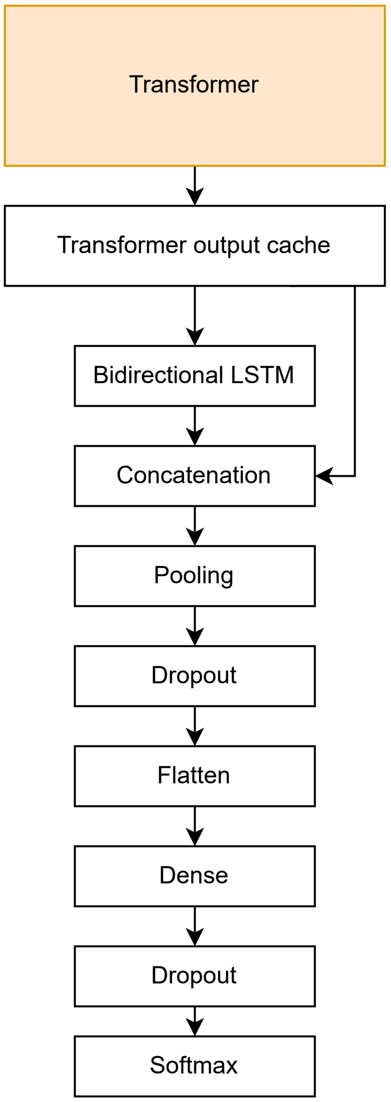

Figure 1: The architecture of the neural network.

For all tested models, the results on the Slovene data were significantly better than on the English data (ranging from 11 % to 18 % improvement). This means that in the used Slovene dataset, sarcasm was more clearly expressed. The best results on both datasets were achieved using the RoBERTa transformer with an embedding length of 100. Even when using shorter embeddings, the RoBERTa transformer performed the best. However, the difference between embeddings of length 100 and 20 was small (only 2 % difference in F1 score). Additionally, the difference between using RoBERTa and DistilBERT transformer is also relatively small (3 % to 6 % difference in F1 score), which implies that the usage of DistilBERT can be a good alternative to RoBERTa on low-cost hardware. When using a multilingual transformer, the results on the English dataset were close to the English-only transformer. However, on the Slovene dataset, the multilingual dataset provided slightly better results.

Table 2: Results of the experiment. The first three columns contain *transformer's name* (**M**), *used dataset* (**L**), and *embeddings length* (**EL**). The dataset is denoted by its language (*slo* for Slovene and *en* for English). In the last four columns, *accuracy* (**A**), *precision* (**P**), *recall* (**R**), and *F1 norm* (**F1**) are presented. The results are rounded to two decimal places.

M	L	EL	A	P	R	F1
RoBERTa	en	100	0.70	0.67	0.78	0.72
RoBERTa	en	20	0.66	0.63	0.79	0.70
DistilBert	en	20	0.63	0.60	0.74	0.67
DistilBert - mult	en	20	0.61	0.58	0.80	0.67
RoBERTa	slo	100	0.83	0.82	0.95	0.88
RoBERTa	slo	20	0.81	0.83	0.89	0.86
DistilBert	slo	20	0.74	0.83	0.78	0.80
DistilBert - mult	slo	20	0.79	0.89	0.78	0.83

In [13], the performance of the RCNN-RoBERTa model was measured on various datasets. The F1 score was between 78 % and 90 %, which is considerably better than the results in English, but comparable to the Slovene dataset. However, since different datasets were used in the study, the results are hard to compare. The same English dataset as applied here was used in [11], where participants presented 13 different solutions, ranging between an F1 score of 0.58 and 0.83 Only three solutions were better than the F1 score of 0.72 , which we achieved with RoBERTa transformer and embeddings length of 100

Dataset construction is one of the most critical parts of the experiment since it provides the knowledge base for the transformer models. According to the obtained results, models were able to detect sarcasm in the given examples, despite several drawbacks explained in section 3. The used approach is good enough for uncomplicated use cases where sarcasm is meant to be detected since users also annotate messages with #sarcasm. But for more complicated use cases with complex and more challenging examples, alternative methods to the dataset construction should also be explored.

6 Conclusion

In this paper, we compare how the utilization of different transformers combined with the BiLSTM model affects the accuracy of sarcasm prediction. RoBERTa, English-based DistilBERT, and multilingual DistilBERT transformers were used in the experiment. All three transformers were combined with the same BiLSTM model and trained on English and Slovene datasets. Afterwards, the accuracy of the models was obtained using test datasets in English and Slovene language. In the best case, F1 scores of 0.72 and 0.88 were achieved on English and Slovene datasets, respectively.

In the future, more work could be done on dataset creation. Different dataset construction approaches, as described in section 3, can be explored and adjusted for the Slovene language. Furthermore, current datasets can be expanded by adding more Tweets (especially Slovene) or data from different sources, e.g., Reddit. Another interesting direction for future work is exploring transfer learning to reuse models on different languages, e.g., languages similar to Slovene.

References

[1] G. G. Chowdhury. Natural language processing. *Annual Review of Information Science and Technology*, 37(1):51–89, 2003. DOI: 10.1002/aris.1440370103.

[2] A. Joshi, P. Bhattacharyya, and M. J. Carman. Automatic sarcasm detection: a survey. *ACM Comput. Surv.*, 50(5), Sept. 2017. DOI: 10.1145/3124420.

[3] J. Tepperman, D. Traum, and S. Narayanan. "Yeah Right": sarcasm recognition for spoken dialogue systems. In *Ninth international conference on spoken language processing*, 2006.

[4] T. Veale and Y. Hao. Detecting ironic intent in creative comparisons. In *Proceedings of 19th European Conference on Artificial Intelligence - ECAI 2010*, pages 765–770, Amsterdam, The Netherlands. IOS Press, 2010. DOI: 10.3233/978-1-60750-606-5-765.

[5] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020. DOI: 10.1007/s11431-020-1647-3.

[6] S. Amir, B. C. Wallace, H. Lyu, P. Carvalho, and M. J. Silva. Modelling context with user embeddings for sarcasm detection in social media. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 167–177, Berlin, Germany. Association for Computational Linguistics, Aug. 2016. DOI: 10.18653/v1/K16-1017.

[7] R. Kulshrestha. Transformers. 2020. URL: https://towardsdatascience.com/transformers-89034557de14 (visited on 06/20/2021).

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018. arXiv: 1810.04805.

[9] 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, 2019. arXiv: 1907.11692.

[10] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019. arXiv: 1910.01108.

[11] D. Ghosh, A. Vajpayee, and S. Muresan. A report on the 2020 sarcasm detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 1–11, Online. Association for Computational Linguistics, July 2020. DOI: 10.18653/v1/2020.figlang-1.1.

[12] H. Srivastava, V. Varshney, S. Kumari, and S. Srivastava. A Novel Hierarchical BERT Architecture for Sarcasm Detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 93–97, Stroudsburg, PA, USA. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.figlang-1.14.

[13] R. A. Potamias, G. Siolas, and A. G. Stafylopatis. A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, 32(23):17309–17320, Dec. 2020. DOI: 10.1007/s00521-020-05102-3.

[14] A. Khatri and P. P. Sarcasm detection in tweets with BERT and GloVe embeddings. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 56–60, Online. Association for Computational Linguistics, July 2020. DOI: 10.18653/v1/2020.figlang-1.7.

[15] M. Bouazizi and T. Otsuki Ohtsuki. A pattern-based approach for sarcasm detection on twitter. *IEEE Access*, 4:5477–5488, 2016. DOI: 10.1109/ACCESS.2016.2594194.

[16] M. Khodak, N. Saunshi, and K. Vodrahalli. A Large Self-Annotated Corpus for Sarcasm, 2017. eprint: 1704.05579.

[17] R. Misra and P. Arora. Sarcasm Detection using Hybrid Neural Network, 2019. arXiv: 1908.07414.

[18] L. Famiglini. Irony-Sarcasm-Detection-Task. URL: https://github.com/lorenzofamiglini/Irony-Sarcasm-Detection-Task (visited on 06/25/2021).

[19] Google Colab. URL: https://colab.research.google.com/notebooks/intro.ipynb (visited on 06/06/2020).