Performance Evaluation of Pre-trained Models in Sarcasm Detection Task

Haiyang Wang, Xin Song, Bin Zhou(✉), Ye Wang, Liqun Gao, and Yan Jia

National University of Defense Technology, ChangSha, China
{wanghaiyang19,songxin,binzhou,ye.wang}@nudt.edu.cn,
jiayanjy@vip.sina.com

Abstract. Sarcasm is a widespread phenomenon in social media such as Twitter or Instagram. As a critical task of Natural Language Processing (NLP), sarcasm detection plays an important role in many domains of semantic analysis, such as stance detection and sentiment analysis. Recently, pre-trained models (PTMs) on large unlabelled corpora have shown excellent performance in various tasks of NLP. PTMs have learned universal language representations and can help researchers avoid training a model from scratch. The goal of our paper is to evaluate the performance of various PTMs in the sarcasm detection task. We evaluate and analyse the performance of several representative PTMs on four well-known sarcasm detection datasets. The experimental results indicate that RoBERTa outperforms other PTMs and it is also better than the best baseline in three datasets. DistilBERT is the best choice for sarcasm detection task when computing resources are limited. However, XLNet may not be suitable for sarcasm detection task. In addition, we implement detailed grid search for four hyperparameters to investigate their impact on PTMs. The results show that learning rate is the most important hyperparameter. Furthermore, we also conduct error analysis by means of several sarcastic sentences to explore the reasons of detection failures, which provides instructive ideas for future research.

Keywords: Sarcasm detection · Pre-trained models · Natural language processing

1 Introduction

With the thriving of social media platforms such as Twitter and Instagram, the exchange of opinions and ideas among people becomes more frequent than ever. The rise of communication among people stimulates the use of figurative and creative language including sarcasm or irony. Sarcasm usually occurs when there is some discrepancy between the literal and the intended meaning of a text. Consider the following text: *yeah! It is so great to be able to work all day on weekends.* This sentence contains *yeah* and *great*, which seems to express the happiness of the author but actually expresses the negative and complaining sentiment of the overworked weekends. Besides, sarcasm can be manifested in

W. Zhang et al. (Eds.): WISE 2021, LNCS 13081, pp. 67–75, 2021.
https://doi.org/10.1007/978-3-030-91560-5_5

many implicit and concise ways so that Oscar Wilde [3] described sarcasm as "the lowest form of wit but the highest form of intelligence."

It is common knowledge that the goal of Natural Language Processing (NLP) is to understand and generate human language. The sarcasm detection task is of great significance in NLP tasks due to sarcasm can express different meanings from the literal. Moreover, sarcasm detection has great potential in the domains of semantic analysis, such as stance detection and sentiment analysis. However, sarcasm detection is a very difficult task. Firstly, the expression of sarcasm is influenced by many factors such as culture, author's background and conversation context etc. Then, the effectiveness of sarcasm detection models depends on the availability and quality of labelled data used for training. However, collecting such data is challenging due to the subjective nature of sarcasm.

Recently, substantial research has shown that pre-trained models (PTMs) can learn universal language representations. Such excellent representations may be beneficial for overcoming the difficulties of sarcasm detection and can avoid training a new model from scratch. Therefore, we consider the following questions: (1) Can PTMs help improve the performance of the sarcasm detection task? (2) How different PTMs perform in sarcasm detection tasks? (3) How do different hyperparameters of PTMs affect the performance of sarcasm detection?

In order to explore the performance of PTMs in sarcasm detection task, we evaluate five models: BERT [2], RoBERTa [8], DistilBERT [11], ALBERT [6] and XLNet [12]. All models are pre-trained on large unlabelled corpora by applying self-supervised objective. We develop a detailed grid search for four hyperparameters to investigate the function and effectiveness of various hyperparameters on four well-known Twitter sarcasm detection datasets which are SemEval [5], iSarcasm [9], Ptacek [10]and Ghosh [3]. SemEval and Ptacek datasets are labelled by distant supervision where texts are considered sarcastic if they meet predefined criteria, such as including specific hashtags (*#sarcasm*). The datasets of iSarcasm and Ghosh are labelled by tweets authors.

Our contributions are summarized as follows:

- We evaluate five PTMs comparing with the state-of-the-art baselines in sarcasm detection task. According to the results of comprehensive experiments, we analyse the performance difference of PTMs.
- We implement a detailed grid search for four hyperparameters. Based on the results, we give some constructive and meaningful suggestions for the hyperparameter selection of PTMs in the sarcasm detection task.
- We employ four well-known and up-to-date sarcasm detection datasets and introduce the characteristics of different datasets in detail. We also perform error analysis to better explain the power and limitation of PTMs.

2 Related Work

The methods of sarcasm detection in recent research work can be divided into three categories, machine learning methods based on feature engineering [5], conventional deep learning methods [7], and PTMs [1]. Moreover, the models

can also be divided into single text-based or context-based according to the information used in the detection [9].

3 Methods and Datasets

3.1 Pre-trained Models

With the development of deep learning, various PTMs have been widely used to solve NLP tasks and yield outstanding performance.

BERT is designed to pretrain deep bidirectional representations from unlabelled text by conditioning simultaneously the possibility of both left and right word context in all layers [2]. It applies the combination of MLM and NSP as a pre-training objective.

RoBERTa is pre-trained with larger batches of longer sequences for longer time on over 160GB unlabelled text corpora [8]. It only uses MLM as the training objective after comprehensive comparative experiments.

DistilBERT is a distilled version of BERT [11]. It reduces the size of BERT model while retaining 97% language understanding capabilities by a leverage distinct knowledge distillation approach.

ALBERT is a lite BERT that has significantly fewer parameters than traditional BERT [6]. It combines two parameter reduction techniques which are the factorized embedding parameterization and the cross-layer parameter sharing technique. It leverages a self-supervised loss for sentence-order prediction to further improve the performance.

XLNet is a generalized autoregressive pre-trained method [12]. It leverages the best of both autoregressive (AR) language modeling and autoencoding (AE) while avoiding their limitations. It employs a PLM objective to capture dependency structures among tokens better and eliminate the independence assumption.

In summary, PTMs shows that a very deep model can significantly improve the performance of NLP tasks and can be pre-trained from unlabelled datasets. In this paper, the PTMs are all implemented in PyTorch using the huggingface transformers library. They are finetuned using the Adam algorithm in Ubuntu 18.04.5 LTS with 72 CPUs and 6 GPUs.

3.2 Datasets

We apply four well-known sarcasm datasets of English tweets.

SemEval: It is an irony detection dataset in English tweets [5]. The ironic tweets were collected using irony-related hashtags and were subsequently manually annotated to minimise the amount of noise in the corpus.

iSarcasm: The iSarcasm [9] is a sarcasm dataset labelled for sarcasm directly by their authors. It is created by asking Twitter users to provide both sarcastic and non-sarcastic tweets that they had posted in the past.

Ptacek: Ptacek et al.[10] created a Czech and English tweets dataset for sarcasm detection. In this paper, we only use English tweets. Same as the dataset of SemEval2018 Task 3, they also collected 780,000 English tweets with *#sarcasm* hashtag.

Ghosh: Ghosh and Veale created a self-annotations tweets dataset for sarcasm detection[3]. They used a Twitterbot named @onlinesarcasm. Twitterbot retweets a chosen unlabelled tweet to its author, appending a yes/no question as a comment to elicit a reply. Then the tweet will be labelled according to the author's reply.

4 Experiments Results

In this section, we conduct comparative experiments, detailed grid search and error analysis research to explain the power and limitation of PTMs in sarcasm detection task.

4.1 Comparative Experiments

The goal of the comparative experiments is to evaluate the general performance of PTMs. We evaluate PTMs on four sarcasm datasets and compare five PTMs with the best baseline of each dataset. Table 1 shows the F1-score results.

Table 1. The results of comparative experiments on four sarcasm datasets

Model	SemEval	iSarcasm	Ptacek	Ghosh
Best-Baseline	0.724 [5]	0.364 [9]	0.874 [9]	**0.900** [3]
BERT	0.700	0.654	0.864	0.828
RoBERTa	**0.731**	**0.668**	**0.882**	0.849
DistilBERT	0.678	0.617	0.849	0.846
ALBERT	0.679	0.541	0.856	0.792
XLNet	0.655	0.455	0.872	0.829

As can be observed from the Table 1, RoBERTa performs best on the four sarcasm detection datasets compared to the other four pre-trained models. That's probably because RoBERTa pre-trained in a larger corpus which may contain more ironic texts. In addition, RoBERTa may learn better representations to capture the implicit semantics of texts. BERT performs better than DistilBERT, ALBERT, and XLNet on the SemEval and iSarcasm datasets. For the two light versions of BERT, DistilBERT and ALBERT, they show competitive performance in SemEval and Ptacek. The XLNet model does not show superior performance. Especially on the iSarcasm dataset, XLNet may fall into the trap of local maximum. More detailed explanation can be found in the Sect. 4.2 .

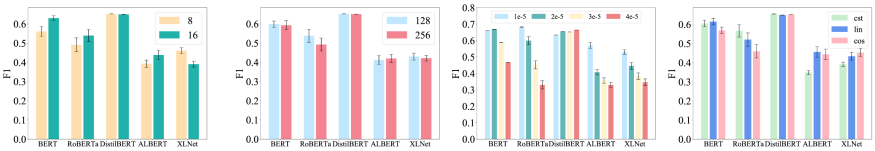
4.2 Grid Search

The fine-tuning of PTMs is similar to transfer learning but different from training the model from scratch. Thus, the experience accumulated in traditional hyperparameter optimization research may not be suitable for the settings of fine-tuning[4]. Our goal is to research the effectiveness and necessity of hyperparameter tuning for fine-tuning in the sarcasm detection task. Thus, we apply a grid search for the hyperparameters: Batch size, Sequence length, Learning rate and Learning rate schedule, which are shown in Table 2 .

Table 2. Search space over chosen hyperparameters.

Hyperparameter	Considered Configurations
Batch size	8; 16
Sequence length	128; 256
Learning rate	1e–05; 2e–05; 3e–05; 4e–05
Learning rate schedule	constant (cst);linear (lin);cosine (cos)

As we all know, the learning rate is the most common and important hyperparameter when fine-tuning. Compared to a model trained from scratch, setting a smaller learning rate during fine-tuning is a better choice. The pre-training model has learned fairly good representations in large-scale corpora during the pre-training process. Excellent representations are very helpful for downstream tasks. A small learning rate can help models make full use of the already gained semantic knowledge. Therefore, we consider learning rate values from 1e-5 to 4e-5. Moreover, we apply three different learning rate schedules. For batch size, we choose 8 and 16 after considering memory limitations and evaluate sequence lengths of 128 and 256 for the grid search.



(a) Batch size (b) Sequence length (c) Learning rate (d) LR schedule

Fig. 1. Average of F1-score on SemEval dataset.

Given the defined hyperparameters in Table 2. There are 48 combinations evaluated for each model and dataset. We analyse the performance of each combination vary with learning rate and the results are shown in Table 3. According

to the F1-score, we can get two different types of configurations which are winning and optimum. (1) **Winning configurations:** Given the learning rate, winning configurations are the hyperparameter combinations in which each model achieves the highest F1-score under each dataset. (2) **Optimum configurations:** Given the dataset and pre-trained model, optimum configurations are the hyperparameter combinations which achieve the highest F1-score. In total, there are 80 winning configurations and 20 optimum configurations in Table 3. Moreover, in order to visually observe the impact of different hyperparameters on performance, we calculate the average F1-score and variance on the dataset of SemEval, as shown in the Fig. 1. We select configurations that have specified hyperparameters and calculate the average of the F1-score.

Table 3. The results of the grid search vary with the Learning Rate (LR). Winner is the winning configuration out of the 12 possible configurations per LR. The format is (Batch size, Sequence length, Learning rate schedule). The F1-score of optimum configurations are indicated in bold. The highest F1-score on each dataset is shown in red.

		BERT		RoBERTa		DistilBERT		ALBERT		XLNet	
	LR	Winner	F1	Winner	F1	Winner	F1	Winner	F1	Winner	F1
SemEval	1e−5	(8,256,cos)	0.6784	(8,256,cst)	**0.7314**	(16,256,cst)	0.6586	(8,256,lin)	**0.6794**	(16,256,cos)	0.6178
	2e-5	(16,256,cst)	**0.6999**	(8,256,lin)	0.7204	(8,128,cst)	**0.6783**	(16,128,lin)	0.6331	(8,128,cst)	**0.6545**
	3e-5	(16,128,lin)	0.6928	(16,128,lin)	0.7104	(16,128,lin)	0.6629	(16,128,cos)	0.6677	(16,128,cos)	0.6066
	4e-5	(8,256, cos)	0.6745	(16,128,lin)	0.3763	(16,128,lin)	0.6758	(16,128,cos)	0.3763	(8,128,cst)	0.3763
iSarcasm	1e-5	(8,128,cst)	0.6334	(8,128,lin)	0.6539	(16,128,cst)	0.5114	(8,128,cos)	0.5265	(8,128,cst)	**0.4546**
	2e-5	(8,256,lin)	**0.6539**	(16,256,lin)	**0.6682**	(16,256,cst)	0.5959	(16,128,cos)	**0.5410**	(8,128,cst)	0.4546
	3e-5	(8,128,cst)	0.6364	(8,128,cst)	0.4546	(16,256,cst)	0.5815	(16,128,cos)	0.4714	(8,128,cst)	0.4546
	4e-5	(16,256,lin)	0.6104	(8,128,cst)	0.4546	(16,128,lin)	**0.6166**	(16,256,lin)	0.5234	(8,128,cst)	0.4546
Ptacek	1e-5	(8,256,cos)	0.8630	(16,128,cos)	**0.8819**	(8,128,cos)	0.8462	(16,128,lin)	**0.8559**	(16,256,lin)	**0.8724**
	2e-5	(16,128,cos)	**0.8636**	(16,256,lin)	0.8813	(8,128,lin)	**0.8486**	(16,128,cos)	0.8517	(16,256,cos)	0.8706
	3e-5	(16,128,cos)	0.8617	(16,128,cos)	0.8752	(16,128,lin)	0.8470	(16,128,cst)	0.7999	(16,256,lin)	0.8699
	4e-5	(16,128,lin)	0.8611	(16,128,cos)	0.8685	(8,256,lin)	0.8451	(8,128,cos)	0.7759	(16,128,lin)	0.8598
Ghosh	1e-5	(8,256,cos)	0.8231	(16,256,cst)	0.8353	(16,256,cos)	0.8161	(8,256,cos)	**0.7921**	(16,128,cst)	**0.8294**
	2e-5	(16,128,lin)	**0.8281**	(16,128,cst)	**0.8485**	(16,128,cst)	0.8407	(16,256,lin)	0.7644	(16,256,cos)	0.8051
	3e-5	(16,128,lin)	0.8226	(16,256,cos)	0.8198	(16,256,cos)	0.8164	(8,128,cst)	0.3305	(16,128,cos)	0.7700
	4e-5	(16,128,lin)	0.7939	(16,256,cos)	0.7946	(16,128,cst)	**0.8455**	(16,256,lin)	0.7539	(8,128,cst)	0.3361

Next, we discuss the results of grid search according to the types of hyperparameters.

Batch Size. The best-performing batch is 16 whether it is considered globally or from the 20 highest F1-scores. Related research also shows that larger batches could help achieve more accurate gradient estimation to some extent. On the contrary, the small batch may cause additional noise during model training. However, we also find that the model with a lower learning rate and smaller batch is more likely to achieve better performance.

Sequence Length. Intuitively from Fig. 1(b), a sequence length of 256 does not lead to better performance. For the RoBERTa, the longer sequence length makes the average F1-score decrease greatly. According to the results of grid search,

there is an indication that short sequence length may help PTMs achieve better performance on tweets datasets.

Learning Rate. In general, the best performing learning rate is 1e–5 and 2e–5. From Table 3, we can see that updating the pre-trained parameters by a small learning rate is important to maximize the performance of PTMs. Further more, we can see that from Fig. 1(c). Only DistilBERT model has a slight ascent with the increase of learning rate. BERT, RoBERTa, ALBERT and XLNet all have huge performance declines when the learning rates are 3e-5 and 4e-5. Considering the reason, DistilBERT use knowledge distillation approach, which may help it become more robust.

Learning Rate Schedule. Evaluating the learning rate schedule, the number of winning configuration with the constant, the linear or the cosine schedule is very close which are 25, 27 and 28 respectively. As for optimum configurations, the linear and constant schedule are chosen frequently. As shown in Fig. 1(d), different PTMs have various learning rate schedule preferences.

Comprehensive consideration of overall performance, RoBERTa achieves excellent performance in the sarcasm detection task. Surprisingly, DistilBERT outperforms BERT on the Ghosh dataset. It also outperforms XLNet on the SemEval, iSarcasm and Ghosh datasets. However, ALBERT and XLNet do not perform as well as expected. The performance of ALBERT still lags behind that of BERT despite the use of large variants. As shown in Table 3 , XLNet falls into the trap of local maxima on iSarcasm dataset. The first reason is the unbalanced distribution of labels in test set. Besides, XLNet uses the PLM as the training objective which can help XLNet capture dependencies between tokens. However, sarcasm detection belongs to segment-level task [4] while PLM may not help XLNet achieve better performance.

4.3 Error Analysis

In this section, we analyse two typical error instances. These instances are all sarcastic tweets while the model judges them to be non-sarcastic tweets. However, it's easy for humans to recognize them as sarcasm.

Sometimes I feel maths is the only place in this world that really accepts me. Numbers are my life.

The author of this tweet expresses his love for mathematics and his disappointment with the real world. The irony of this sentence is aimed at the author himself. The reason why this tweet is wrongly classified may be that there are no obvious positive and negative words in the text at the same time. Sarcastic statements are often characterised by a form of opposition or contrast. In this tweet, the form of opposition is very cryptic.

I love when my parents yell and scream at me and then get pissed at me for crying.

This is an ironic tweet. The author expressed dissatisfaction with his/her parents. The misclassification of tweets may be due to the corpora of PTMs. In large unlabelled corpora, the text when *love* and *parents* appear together often express a positive emotion and true love to parents. A pre-trained model fine-tuned with a little sarcasm data cannot capture the implicit meaning of the tweet. Therefore, the pre-trained model is likely to make some preconceived judgements due to the impact of the corpora.

5 Conclusion

In this paper, we evaluate the performance of various PTMs on the four well-known sarcasm detection datasets. Experimental results show that the RoBERTa model performs strongly on several sarcasm detection datasets and exceeds the best baseline. Furthermore, the BERT-based approaches outperform XLNet on several datasets. Then we summarize some rules about the hyperparameter selection of PTMs for the sarcasm detection task. Smaller learning rate, shorter sequence length and larger batch size can help PTMs achieve better performance in detecting sarcastic tweets task. In the future, we plan to investigate sarcasm detection with contextual information and explore whether pre-training on large unlabelled corpora cause PTMs to make some biased judgments.

Acknowledgements. This work was supported by the National Key Research and Development Program of China No. 2018YFC0831703.

References

1. Baruah, A., Das, K., Barbhuiya, F.A., Dey, K.: Context-aware sarcasm detection using bert. In: Fig-Lang@ACL (2020)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
3. Ghosh, A., Veale, T.: Magnets for sarcasm: making sarcasm detection timely, contextual and very personal. In: EMNLP (2017)
4. Guderlei, M., Aßenmacher, M.: Evaluating unsupervised representation learning for detecting stances of fake news. In: COLING (2020)
5. Hee, C.V., Lefever, E., Hoste, V.: Semeval-2018 task 3: irony detection in english tweets. In: SemEvalNAACL-HLT (2018)
6. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: a lite bert for self-supervised learning of language representations. ArXiv (2020)
7. Lemmens, J., Burtenshaw, B., Lotfi, E., Markov, I., Daelemans, W.: Sarcasm detection using an ensemble approach. In: Fig-Lang@ACL (2020)
8. Liu, Y., et al.: Roberta: a robustly optimized bert pretraining approach (2019). ArXiv abs/1907.11692
9. Oprea, S., Magdy, W.: isarcasm: a dataset of intended sarcasm (2020). ArXiv abs/1911.03123
10. Ptácek, T., Habernal, I., Hong, J.: Sarcasm detection on Czech and English twitter. In: COLING (2014)

11. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. ArXiv (2019)
12. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: Xlnet: generalized autoregressive pretraining for language understanding. In: NeurIPS (2019)