REPRODUCE BERTSCORE: EVALUATING TEXT GENERATION WITH BERT

Yu Zhang, Ming He, Yichu Wang, Running Wang
School of Electronics and Computer Science
University of Southampton, UK
$\{yz8n21, mh9n21, yw5n21, rw3n20\}$ @soton.ac.uk

ABSTRACT

The BERTScore is one novel automatic metric to evaluate the correlation of the natural language text generation. The similarity metric is estimated between tokens in the candidate sentence and those in the reference sentence from Zhang et al. (2019). In this paper, BERTScore is re-implemented to assess its correlation and reproducibility. From WMT18 dataset[1], the data of news commentary from diverse languages would be trained into chosen models. Meanwhile, from evaluating scores of each experience, it has been verified that BERTScore is better than other existing metrics. Eventually, all codes and other information would be shown in the GitHub[2].

1 INTRODUCTION

The most common existing method to automatically evaluate the natural language test generation is implementing the comparison between the candidate sentence and annotated references. It can efficiently appraise how outputs from the text generation differ from the human judgement using the computed score from models. From Zhang et al. (2019), the BERTScore metric has been declared to estimate the similarity of two sentences through computing sums of the cosine similarities between the embedding of their tokens. Meanwhile, from several experiences, the model with stronger performance also can be selected based on the BERTScore. In this paper, this result will be accordingly reproduced. Each significant portion will be illustrated and implemented well.

1.1 BERT-SCORE

In the BERTScore, all generated texts by pre-trained BERT contextual embedding from Devlin et al. (2018) would be evaluated. Its score results are obtained by computing degrees of resemblances for each token in the candidate sentence x and the reference sentence $\hat{\mathbf{x}}$. In the computation, each token is expressed by the contextual embedding. Additionally, the matching degree is calculated through the cosine similarity. The detail computation process is announced below.

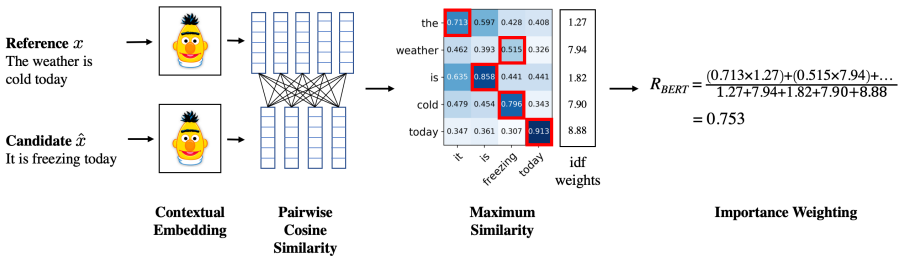

Figure 1: The computation process for BERTScoreZhang et al. (2019)

[1]Website: https://statmt.org/wmt18/translation-task.html#download

[2]Website: https://github.com/soton2022DLgroup/BERTSCORE-EVALUATING-TEXT-GENERATION-WITH-BERT-COMP6248

As the process shown above, the BERTScore would be measured from three aspects, such as precision(P), recall(R) and F1(F) showing in Figure 2. The precision is estimated by matching tokens in $\hat{\mathbf{x}}$ to those in x. Furthermore, the recall is estimated by matching tokens in x to those in $\hat{\mathbf{x}}$. Afterwards, these scores of resemblances are ascended by utilizing the greedy matching. By integrating the precision and recall, F1 is computed to illustrate the comprehensive performance for the model.

$$R_{\text{BERT}} = \frac{1}{|x|}\sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \quad , \quad P_{\text{BERT}} = \frac{1}{|\hat{x}|}\sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \quad , \quad F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}$$

Figure 2: The calculation function for P, R, F from Zhang et al. (2019)

1.2 PRE-TRAINED MODEL

In this re-implementation, four varieties pre-trained model would be utilized, such as BERT, RoBERTa and XLMs.

- BERT model: By Devlin et al. (2018), BERT is a transformers model pre-trained on a large corpus of English data in a self-supervised environment. That means that it is pre-trained on the raw data only, without any human labels.
- RoBERTa model: As the description of Liu et al. (2019), RoBERTa is the improved model on the basis of BERT. By comparing with BERT, it was pre-trained with the Masked language modeling (MLM) objective. When taking a sentence, the model randomly masks 15% of the words in the input, before running the entire masked sentence through the model and predicting the masked words. It leads that this model can learn a bidirectional representation of the sentence.
- XLNet model: In Yang et al. (2019), the XLNet a autoregressive model leveraging the benefits of ARAutoregressive) and AE (Autoencoder) to enhance the performance of BERT. According to the combination and the maximized log likelihood in each sequence, this model can ascend the capturing information from the bidirectional context. Meanwhile, by the contribution of AR, the XLNet does not depend on the data corruption, which improve the robustness of XLNet.
- XLMs model: From Lample & Conneau (2019), Cross-lingual language models (XLMs) is a transformer language models trained unsupervised on a large text corpus and fine-tuned supervised on natural language understanding (NLU) tasks. Unlike the previous works, which are largely English-centric, this model mainly focuses on cross-lingual sentence representations in many languages.

2 IMPLEMENTATION DETAILS

2.1 ADVANCE PREPARATION

The Python version should be greater than or equal to 3.6 and the PyTorch version should be greater than or equal to 1.0.0. Afterwards, the bert_score and transformers packages should be installed. Finally, the GPU need to check whether it is enough or not. In this experiment, the GPU is Tesla P100-PCIE.

2.2 IMPLEMENTATION DESIGN

During designing the implementation, after several tries, the reason why small dataset from WMT18 is processed in training is that the computer performance constraints do not allow to implement large orders of magnitude. Therefore, the news commentary is chosen, which is composed of monolingual training data of diverse language, including Czech(cs), German(de), English(en), Russian(ru), and Chinese(zh). The size of the overall language data is 92MB. The largest length of the input data is 464949(en), which is the suitable length in existing GPU. Except English, other languages should be translated to English. Afterwards, the translated result will compare with the

original English commentary to estimate the similarity.

After determining the data, 9 pre-trained models are utilized, as mentioned in introduction, to compare the performance of different models. These pre-trained models respectively are Bert-base-uncased, Bert-large-uncased, Bert-base-cased-mrpc, Roberta-base, Roberta-large, XLNet-base-cased, XLNet-base-large, XLM-mlm-en-2048 and XLM-mlm-100-1280. It is meaningful to compute the better model in the text generation for diverse language.

Based on these prerequisites, all training processes would be implemented in turn. After implementing, the precision, recall and F1 measure will be estimated in mean to evaluate different language generation tasks.

3 RESULT

In this section, experiment results would be illustrated and compared with the provided result from Zhang et al. (2019). These results estimate the Kendall correlations with segment-level human judgments. Precision, Recall and F1 scores for each tasks are shown in Table 1. From other language-to-English translation task, Robert-large and XLM-mlm-en-2048 model has the best performance. Bert-base-cased-mrpc model has the second best performance. With comparing to the result in original paper in Table 2, the performance of the implementation on smaller dataset is not as good as that on WMT18 on some tasks, like de-en translation. However, for cs-en task and ru-en task, most of our performance is better as marked in Table 1. XLM model performs better in all tasks.

Table 1: Results of different models on smaller dataset

Models	cs-en			de-en			ru-en			zh-en		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Robert-base	**0.712**	**0.785**	**0.746**	**0.744**	**0.800**	**0.771**	**0.715**	**0.776**	**0.744**	**0.722**	**0.765**	**0.743**
Robert-large	**0.704**	**0.833**	**0.763**	**0.741**	**0.837**	**0.786**	**0.654**	**0.827**	**0.730**	**0.697**	**0.824**	**0.755**
Bert-base-uncased	0.272	0.300	0.284	0.262	0.311	0.283	0.180	0.276	0.216	0.190	0.279	0.224
Bert-large-uncased	0.355	0.403	0.375	0.361	0.408	0.381	0.262	0.388	0.311	0.281	0.346	0.309
Bert-base-cased-mrpc	**0.437**	**0.506**	**0.468**	0.436	0.510	0.469	0.364	0.488	0.415	0.362	0.486	0.413
XLM-mlm-100-1280	**0.419**	**0.432**	**0.424**	0.426	0.438	0.431	**0.431**	**0.438**	**0.433**	**0.405**	**0.423**	**0.413**
XLM-mlm-en-2048	**0.770**	**0.807**	**0.788**	**0.766**	**0.804**	**0.785**	**0.752**	**0.753**	**0.752**	**0.806**	**0.774**	**0.789**
XLNet-base	0.281	0.330	0.301	0.275	0.338	0.301	0.241	0.333	0.277	0.318	0.239	0.270
XLNet-large	0.240	0.304	0.267	0.234	0.305	0.264	0.225	0.282	0.249	0.215	0.151	0.176

Table 2: Results of different models presented in the paper Zhang et al. (2019) on WMT18 dataset

Models	cs-en			de-en			ru-en			zh-en		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Robert-base	0.368	0.383	0.391	0.530	0.536	0.540	0.318	0.336	0.339	0.235	0.245	0.249
Robert-large	0.387	0.388	0.404	0.541	0.546	0.550	0.345	0.343	0.353	0.248	0.255	0.264
Bert-base-uncased	0.349	0.370	0.373	0.522	0.538	0.531	0.325	0.333	0.341	0.232	0.244	0.243
Bert-large-uncased	0.361	0.386	0.402	0.529	0.532	0.537	0.340	0.347	0.344	0.241	0.247	0.252
Bert-base-cased-mrpc	0.348	0.379	0.373	0.522	0.531	0.534	0.318	0.339	0.342	0.224	0.242	0.242
Xlm-mlm-100-1280	0.349	0.358	0.358	0.516	0.518	0.525	0.310	0.320	0.322	0.233	0.237	0.238
Xlm-mlm-en-2048	0.355	0.362	0.367	0.527	0.528	0.531	0.311	0.333	0.330	0.238	0.240	0.246
XLNet-base-cased	0.339	0.364	0.355	0.516	0.521	0.524	0.307	0.317	0.320	0.236	0.238	0.241
XLNet-large-cased	0.348	0.366	0.375	0.520	0.529	0.530	0.319	0.331	0.332	0.235	0.241	0.240

4 DISCUSSION

By comparing implementation results and results from the paper, the characteristic of data is similar. The RoBERTa-large and XLM-mlm-en-2048 performs significantly better than other models. For the RoBERTa-large, the reason is that during training, it has the specific objective, which is the Masked language modeling (MLM). It is really different from the recurrent neural networks(RNNs) or some autoregressive models like GPT. It can be effectively learn the bidirectional representation from sentences. Based on this distinction, an inner representation would be learnt by the model, which can enhance the features extraction ability for some downstream tasks. Additionally, for the XLM, this model is mainly centered on cross-lingual sentence representations in many languages. Therefore, they both have the strong performance in these tasks.

In addition, from results of the paper, the BERT-base model performed slightly better than XLNet. Meanwhile, the XLNet also has the lowest performance for all results. These basic characteristics of the implementation was following those of the paper.

Through comparing the differences between BERT and RoBERTa, RoBERTa adopts longer training time, larger batch size. Furthermore, the task of next sense prediction for RoBERTa was abandoned in the pre-training. These strengthen the stability of RoBERTa and enhance the performance for downstream tasks of NLP.

Nevertheless, there is some distinction between implemented results and those results on the WMT18 dataset. For instance, the gap between maximum and minimum is pretty large. Or minimum results for XLNet-large are tiny. The reason is from two aspects. For the initial aspect, the quality of chosen data has an influence on the performance. The news commentary has many special vocabularies. Additionally, there may be words whose comprehension is diverse in various contexts. For the second aspect, the length of data is not enough. Less data may leads that the model cannot learn enough information to generate better text. Especially in previous case, this complex situation should be in need of learning more features.

5 CONCLUSION

There are 9 pre-trained NLP models implemented on English Translation tasks from other four languages(cs, de, ru, zh) to English based on a small subset of the WMT18 dataset. After processing all experiences, precision(P), recall(R) and F1(F) are computed to evaluate each model. The Robert-large and XLM-mlm-en-2048 have the best performance, and Bert-base-cased-mrp follows them. In most cases, the implemented result is worse than that of the original paper. Whereas, for the cs-en task and ru-en task, the accomplished result is better. XML models perform better on all tasks in experiences than in the original paper. The reason is the complex vocabularies and the insufficiency of chosen data.

5.1 FINDINGS

Meanwhile, after this re-implementation, it can proof that this BERTScore computing on each model in this paper is reproducible. Nevertheless, with ascending of the dimensions of data, the computing burden would be also ascending. If candidate sentences and reference sentences have been prepared, the BERTScore would be a better metric. Eventually, there is one issue, which is found in the BERTScore. If using the pre-trained model of BERTScore to do prediction, the memory cannot be utilized efficiently. It would waste too much computation space.

REFERENCES

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.