Applying Text-Classification on Question-Answering: An Attempt to Prove the Performance of the Bert-based Module

Yexin Wu
ACM Class 2019
Shanghai Jiao Tong University
wuyexin_libro_i131@sjtu.edu.cn

Abstract

Bidirectional **E**ncoder **R**epresentations from **T**ransformers(**Bert**) proved an effective model for improving many natural language processing tasks. In this paper, we attempt to improve the performance of Question-Answering(QA) tasks by applying article/paragraph-level text classification to the dataset, which is shown to be helpless. With the help of bertviz, we find out some attention pattern which is possibly associated with QA tasks.

1 Introduction

Fine-tuning the pre-trained model such as Bert[3] is widely applied to the language processing downstream problems, including Named Entity Recognition, Sentence-Classification, and Question-Answering(QA). This paper focuses on promoting the performance of the QA task.
Limited to the computing resource and time, we find it hard to design a different structure and pre-train the model. Hence processing the data is one feasible direction to improve the model. Since articles in different fields have different writing styles and expression choices, dividing the articles/paragraphs into several categories and fine-tuning Bert models respectively on these categories is possible to obtain better models on more granular sub-tasks. By combining these models, we try to form an ensemble system with better accuracy.
Nevertheless, Text-Classification on articles/paragraphs does not improve the performance as the experiment shown. Therefore we dive into the attention mechanism with the help of bertviz, trying to understand how Bert works out QA tasks and why article-level classification can not help improve performance. Our code is available at `https://github.com/LibroWu/TCQA-Bert`.

2 Related work

2.1 Transformers and BERT

Bert is a large network consisting of multi-layer bidirectional Transformer[6] encoders. Each layer contains multiple self-attention heads. An attention head deals with three sequences of vectors: query, key, and value q_i, k_i, v_i. By computing attention weights α between query and elements in the key sequence, the attention outputs the result o with the weight and value.

$$\alpha_{ij} = \frac{exp(q_i^T k_j)}{\sum_{l=1}^{n} \exp(q_i^T k_l)}, o_i = \sum_{j=1}^{n} \alpha_{ij} v_j.$$

A self-attention head is an attention head with one sequence simultaneously serving as the query, key, and value. Introducing the self-attention mechanism into the Transformer contributes to bidirectionally

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

relating the context. Furthermore, multi-head introduces a linear layer to project the three sequences, allowing the learnability of the model. Piled up with the encoders of Transformer, Bert is firstly pre-trained with the Cloze task and "next sentence prediction" task. Then it is applied to different downstream tasks.

2.2 XLNet

XLNet [8] is a kind of AR language model that builds the probability distribution of a text corpus. It (1) enables learning bidirectional contexts by dealing with the permutation language modeling and (2) overcomes the pretrain-finetune discrepancy of Bert.

2.3 Unsupervised Text-Classification with SimBert

The text is manually and unsupervised tagged. Researches show that Bert is unsuitable for unsupervised text classification. There is anisotropy in word vectors because of the influence of word frequency on spatial distribution and sparsity.[5]Hence we use SimBert [4] , a bert-based model specialized for semantic textual similarity (STS) task, to do the classification. Moreover, T-distributed Stochastic Neighbor Embedding (tSNE)[1] is applied to visualize the high-dimension vectors.

2.4 Dataset: SQuAD2.0

The Stanford Question Answering Dataset(SQuAD) is a reading comprehension dataset consisting of questions on a set of Wikipedia articles. The answer to every question is a segment of text from the corresponding reading passage. In version 2.0, the question might be unanswerable. The dataset can be downloaded and explored from `https://rajpurkar.github.io/SQuAD-explorer`.

2.5 Bertviz

BertViz is an interactive tool for visualizing attention in Transformer language models such as BERT, GPT2, or T5. The website (`https://github.com/jessevig/bertviz`) lists the notebooks and tutorials of the tool. We use *model_view_bert.ipynb* to visualize attention layers of Bert.

3 Method

We manually classify the texts into four categories: ArtLang (art and language), HisGeo (history and geography), IntroBio (introduction and biography), and TechSci (technology and science). The criterion is mainly the title of the article. Then models are trained respectively (6 epochs to make up for smaller training data) on these sub-datasets and evaluated on the corresponding dataset (the classification of the test data is based on XLNet and reaches 0.8 accuracy, but due to the bad performance of these sub-models, we do not combine them with a classifier).
Then we use SimBert and K-means to unsupervised tag the data at a paragraph level. Firstly use SimBert to encode the paragraph into a sequence of vectors. Choose the vector of [CLS] to represent the paragraph. Apply k-means to the 768-dimension space and obtain 4 clusters. Divide the dataset according to the result of K-means. Train models, respectively. Randomly choose several paragraphs to form a sub-dataset of a similar size (5000 in our work) to the other 4 clusters. Train the model labeled 'Random' on this dataset to be the baseline.
The performance is evaluated by the *Evaluation Script v2.0* on the official website of the SQuAD.
We use bertviz, written by Jessevig, to explore the attention heads of the Bert and try to find out how Bert works to solve Question-Answering and why article/paragraph level classification does not help.

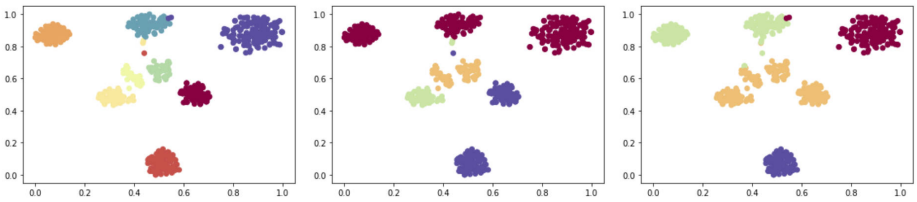
4 Experiments

4.1 Baseline

We use *bert-base-uncased* and *xlnet-base-cased* from hugging face as the baseline. The model based on *bert-base-uncased* is trained for 4 epochs and reaches 71.02 f1. The model based on *xlnet-base-cased* is trained for 3 epochs and reaches 78.62 f1.

4.2 Text classification

The figure 1 illustrates eight articles' distribution after reduction on dimension. The categories and titles of these articles are as follows.
Art and language: *The Legend of Zelda: Twilight Princess* and *Spectre (2015 film)*.
History and geography: *Sino-Tibetan relations during the Ming dynasty*, *2008 Sichuan earthquake* and *New York City*.
Introduction and biography: *Fred Eric Chopin* and *Beyonce*.
Technology and science: *IPod*.



(a) dyed by articles (b) dyed by manual (c) dyed by cluster

Figure 1: tSNE visualizing the paragraph vectors

There is a difference between manual classification and unsupervised classification. The paragraphs in the same article are distributed closes.

4.3 Question answering

Table 1: Bert Models on manually classified text

Models	Matric			
	Exact	F1	HasAns F1	NoAns F1
all (std)	65.88	71.02	67.57	74.45
HisGeo (std)	58.97	64.11	69.78	58.43
HisGeo (lim)	51.62	57.35	68.71	45.97
ArtLang (std)	68.88	74.56	66.41	83.57
ArtLang (lim)	63.48	67.96	55.85	81.34
IntroBio (std)	71.86	75.38	71.51	78.94
IntroBio (lim)	60.87	63.24	52.43	73.15
TechSci (std)	68.16	73.14	65.83	79.89
TechSci (lim)	62.55	66.90	60.95	72.41

Table 1 shows that Bert-based models trained on a limited dataset have no better performance in the corresponding area as expected. The reason might be the more minor training data, wrong classification, or difference among paragraphs in the same article. So we use unsupervised classification to check these three reasons.

Table 2: XLNet Models on unsupervised classified text

Models	Exact	F1	Matric HasAns F1	NoAns F1
all (std)	75.35	78.62	77.36	80.0
cluster0 (rand)	70.81	74.40	71.05	77.93
cluster0 (lim)	70.69	74.91	72.70	77.23
cluster1 (rand)	71.65	75.22	71.52	78.61
cluster1 (lim)	69.38	73.38	73.04	73.69
cluster2 (rand)	71.55	75.32	73.03	77.68
cluster2 (lim)	71.51	76.02	73.75	78.34
cluster3 (rand)	68.28	71.81	71.92	71.68
cluster3 (lim)	66.11	69.54	72.58	66.17

We use unsupervised classification based on SimBert and K-means to tag the text at the paragraph level. Moreover, a reference model is trained on a randomly selected dataset. Table 2 shows that sub-models have no better performance than the random model. So the unsupervised paragraph-level classification does not help even with the similar training data size. Hence it might be the mechanism of Bert that is not suitable for paragraph classification.

4.4 Dive into attention heads

The former researches [7][2] summarised six attention pattern:

- Pattern 1: Attention to next word.
- Pattern 2: Attention to previous word.
- Pattern 3: Attention to identical/related words.
- Pattern 4: Attention to identical/related words in other sentence.
- Pattern 5: Attention to other words predictive of word.
- Pattern 6: Attention to delimiter tokens.

Furthermore, we find other attention patterns such as attention to previous prep. or verb, attention to next/previous token in the same entity, and attention to antonym.

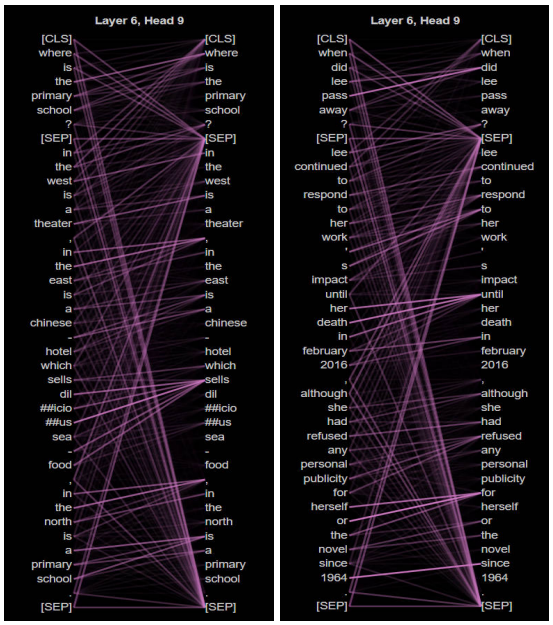


Figure 2: Attention to previous prep. or verb.

Figure 2 shows that *Layer 6, Head 9* projects the word into the nearest preposition or verb previous to it (e.g., "delicious seafood" to "sells" in the left figure and "her death" to "until"). There also exist reversed projects from prepositions or verbs to the nearest entity next to it (these projects might be found in *Layer 7*).

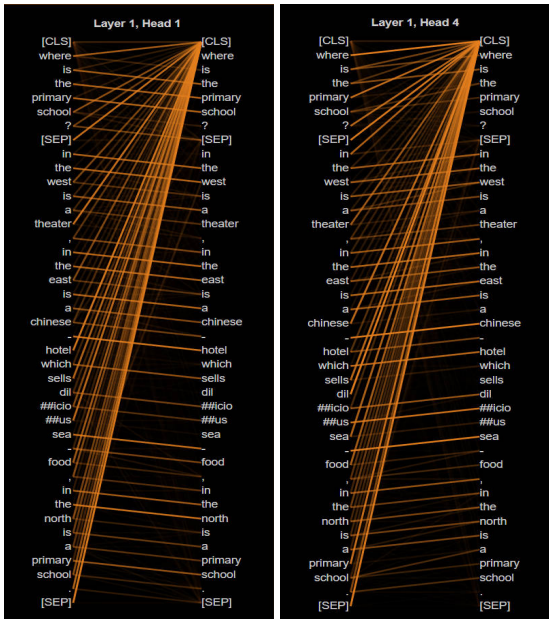


Figure 3: Attention to next/previous token in the same entity

Figure 3 shows the attention to the next/previous token in the same entity. If the word is the end of the entity it belongs to, it is projected to [CLS]. The left figure shows the attention to the next token, and the right shows attention to the previous token.

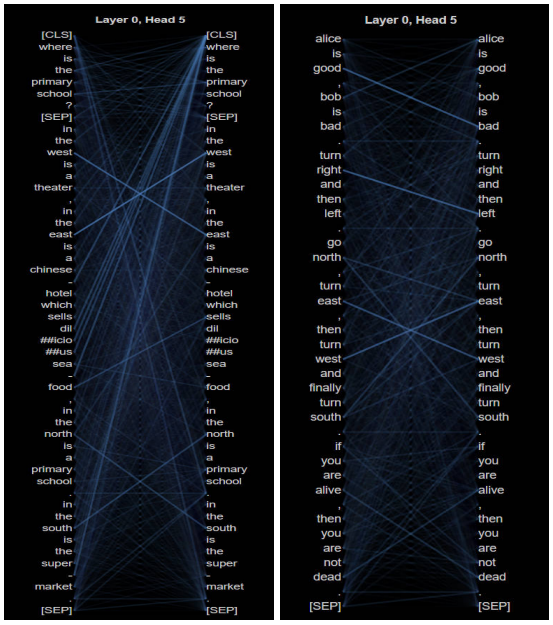


Figure 4: Attention to antonym

Figure 4 shows the attention to antonym. The opposite word pairs are projected mutually, such as "west and east," "north and south," "good and bad," and "dead and alive".
The mechanism of Bert solving the QA task might be as follows. By attention to identical/related words in the other sentence, the answer's prompt in the context is noticed. When negation exists (e.g., not) in either question or context, the attention to antonym is enhanced. If the question is about location or object, the attention to previous/next prep. or verb is considered. When locating the answer span, attention to the token in the same entity is enhanced. These attention patterns mentioned before have little association with the article style or terminology. Therefore, the writing style or terminology has little influence on the Question-Answering.

5 Conclusions and Limitations

The experiment result shows that the classification of paragraphs has no help in improving the performance of the QA task. By analyzing the attention heads of Bert, we find that attention patterns have little association with the article style or terminology. However, the analysis is subjective. Classification accuracy also has a chance to be improved.

References

[1] T. Tony Cai and Rong Ma. *Theoretical Foundations of t-SNE for Visualizing High-Dimensional Clustered Data*. 2022. arXiv: 2105.07536 [stat.ML].

[2] Kevin Clark et al. *What Does BERT Look At? An Analysis of BERT's Attention*. 2019. arXiv: 1906.04341 [cs.CL].

[3] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[4] Tianyu Gao, Xingcheng Yao, and Danqi Chen. *SimCSE: Simple Contrastive Learning of Sentence Embeddings*. 2022. arXiv: 2104.08821 [cs.CL].

[5] Bohan Li et al. *On the Sentence Embeddings from Pre-trained Language Models*. 2020. arXiv: 2011.05864 [cs.CL].

[6] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[7] Jesse Vig. "Deconstructing BERT: Distilling 6 Patterns from 100 Million Parameters". In: (2018). URL: https://towardsdatascience.com/deconstructing-bert-distilling-6-patterns-from-100-million-parameters-b49113672f77.

[8] Zhilin Yang et al. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. 2020. arXiv: 1906.08237 [cs.CL].