BERT-HCMQA: A BERT-Based Model for Hierarchical and Conditional Multi-span Question Answering

Anonymous EMNLP submission

Abstract

Text question answering has been one of the trendiest research topics in machine learning. However, due to the need for professional knowledge, complex context information, multi-span answers and some other requirements, traditional neural network models can hardly handle this task. Fortunately, BERT is good at comprehending the text information. It brings us new opportunities in solving the complex text question answering problem. Based on BERT, we propose BERT-HCMQA, in order to better perform the extractive, hierarchical, and multi-span Question Answering with conditions. The main challenges in developing the model mostly lie in finding several answers to the question simultaneously and then getting the relationships between the answers. To address the challenges, we propose several solutions and our model is currently ranked sixth in the BiendataCCKS 2022 competition up to now. The code and pre-trained models are available at https://github.com/HourunLi/CCKS2022

1 Introduction

Question answering is an important research task in linguistic text analysis. With the development of machine learning models and the release of massive and well annotated datasets from both academic (Rajpurkar et al., 2018; Reddy et al., 2019) and industry (Bajaj et al., 2016; He et al., 2017) communities, continuous success in this area has been made.

For text question answering problems, extractive question answering is an important form, which requires the model to extract text fragments (span) as answers in the corresponding text. Unlike the common single answer (single-span) extraction task, there are many potential answers to a question in real life QA applications, which is also called as multi-answer (multi-span) extractive question answering.

Generally, the extractive and multi-span question answering commonly has the following two features. First, the questions are not clear enough due to the lack of specialized knowledge of the questioners, which leads to the need to answer questions separately according to different conditions. Second, multiple answers to the same question may belong to different granularities, and there may be a potential hierarchical relationship between them. All these elements make it hard to provide an efficient model for hierarchical and conditional multi-span QA problems.

In order to solve the problem, we need to handle the following three main challenges. The first challenge is figuring out how to get the full meaning of the context and understand the terms. The second challenge relates to how to find the potential several answers simultaneously, instead of in the traditional single-span extraction style. The third challenge is how to explore the potential relationships between arbitrary two answers. Fortunately, a powerful tool named BERT provides us with a new method to address the challenges. BERT is a popular attention model making use of bidirectional training of Transformer. Its neural network has massive parameters but is pre-trained, and it can be fine-tuned for specific downstream tasks. Generally, BERT has a deeper sense of language context, and it's a good option for NLP (Devlin et al., 2018).

We propose our model named BERT-HCMQA, based on BERT. It successfully addresses the three challenges. BERT-HCMQA is composed of two main modules, a sequence labeling module and a relation matching module, respectively. The sequence labeling module finds the answers in a context in a sequence labeling way. The Relation matching module matches arbitrary two answers to form all the potential pairs and decides whether there is a specified relation for each pair.

We evaluate BERT-HCMQA on a server with a NVIDIA RTX311 A6000 GPU. Because we are required to design a model to solve the problem in the Biendata competition [1], we do not provide a baseline and benchmark comparison. Our model achieves 92.23% and 99.28% accuracy respectively in the first and second module. Both modules take only 20 minutes to get trained and fine-tuned on the customized datasets. Finally, our model performs the task well and gets the 6th rank up to now.

[1] https://www.biendata.xyz/competition/CMQA/

2 Background

In this section, we introduce the background of our paper.

2.1 Question Answering

Question answering (QA) is a computer science discipline within the fields of information retrieval and natural language processing (NLP), which is concerned with building systems that automatically answer questions posed by humans in a natural language (Cimiano et al., 2014).

There are different QA variants based on the inputs and outputs:

- Extractive QA: The model extracts the answer from a context. The context here could be a provided text, a table or even HTML! This is usually solved with BERT-like models.
- Open Generative QA: The model generates free text directly based on the context. You can learn more about the Text Generation task in its page.
- Closed Generative QA: In this case, no context is provided. The answer is completely generated by a model.

The Extractive QA is the primary focus of this paper. It is difficult to train models that perform these tasks, because models need to understand the structure of the language, have a semantic understanding of the context and the questions, have the ability to locate the position of an answer phrase, and much more(Lewis et al., 2019). Fortunately, the concept of attention in neural networks has been a lifesaver for such difficult tasks. Since its introduction for sequence modeling tasks, lots of RNN networks with sophisticated attention mechanisms like R-NET, FusionNet, etc. have shown great improvement in QA tasks (Xu et al., 2021). However, a completely new neural network architecture based on attention, specifically self-attention, called Transformer, has been the real game-changer in NLP (Tenney et al., 2019).

2.2 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a recent paper published by researchers at Google AI Language (Devlin et al., 2018). It has caused a stir in the Machine Learning community by presenting state-of-the-art results in a wide variety of NLP tasks, including Question Answering (SQuAD v1.1), Natural Language Inference (MNLI), and others.

BERT's key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training (Sarzynska-Wawer et al., 2021; Radford et al., 2018). Devlin et al. (2018) shows that this pre-training of deep bidirectional transformers can have a deeper sense of language context and flow than single-direction language models.

2.3 Pre-training and Fine-tuning

More recently, sentence or document encoders which produce contextual token representations have been pre-trained from unlabeled text and fine-tuned for a supervised downstream task (Lee et al., 2009; Howard and Ruder, 2018). This method is called as pre-traing and fine-tuning. The advantage of this approach is that few parameters need to be learned from scratch.

BERT also takes advantage of this method. And fine-tuning is straightforward since the self attention mechanism in the Transformer allows BERT to model many downstream tasks. Overall pre-training and fine-tuning procedures for BERT are shown in Figure 1. The same pre-trained model parameters are used to initialize models for different downstream tasks. And all parameters are fine-tuned during fine-tuning.

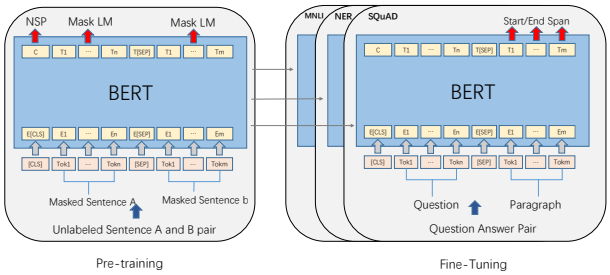

Figure 1: Overall pre-training and fine-tuning procedures for BERT

3 Motivation and Challenges

In this section, we show the motivation of BERT-HCMQA and the major challenges in developing this model.

3.1 Motivation

The traditional sequence model behaves poorly in QA tasks, but the BERT model brings new opportunities to text extractive question answering. First, BERT is a powerful tool to have a deeper sense and comprehensive understanding of context, because it is based on a bidirectional transformer. This really helps us better process the complex context and questions. Second, BERT is pre-trained and can be fine-tuned on customized datasets, which means that we only need to slightly modify the parameters of BERT from scratch and it can achieve excellent performance as well. Third, there are many BERT current downstream task templates. We can further alter the structure and goals of some tasks to build our own model.

Besides, if we can build an excellent model to solve the hierarchical and conditional multi-span question answering problems. It can be applied in so many areas, especially in helping digest massive textual data, developing information services in hospitals and so on (Yoon et al., 2021).

3.2 Challenges

Enabling BERT-HCMQA to tackle the hierarchical and conditional multi-span question answering requires us to handle the following three challenges.

The first challenge comes from trying to get a more comprehensive understanding of the context and question. Because the context generally requires specialized knowledge and overall comprehension, traditional neural networks like RNN can hardly capture the terminology and the overall meaning of a whole and complex context.

The second challenge is how to find the hierarchical answers (conditions, fine answers and coarse answers) at different granularities. First, the answers are multi-span, which means there are several extractive answers for a specific question. It should find the multiple answers simultaneously. Second, answers must vary with different questions even though the context may be the same, which means they must relate to not only the context but the question. However, there is no existing extractive multi-span question answering model meeting these requirements.

The third challenge lies in how to explore the different relations between answers. There may be so many answers generated in the first module, and the answers may be distributed throughout the whole context. Besides, the context may be complex and intricate, which makes it even harder to figure out the relationships.

4 BERT-HCMQA Model

We propose **BERT-HCMQA**, an efficient BERT based model for **H**ierarchical and **C**onditional **M**ulti-span **Q**uestion **A**nswering. In this section, we introduce the general design and the specific modules of our model.

4.1 General Design

Overview. BERT-HCMQA is made up of two major modules: the sequence labeling module and the relation matching module. The first module, the sequence labeling module, based on the BERT token classification task, labels every single word in a sentence. This module works like the NER model but substitutes NER labels with four new labels, i.e. $None, Condition, Fine$ and $Coarse$. By this way, we find out the conditions and coarse or fine answers. The second module, the relation matching module, based on BERT text classification, takes the output of the first module as input source. The second module firstly forms all the potential relations, which generally belong to three types ($condition-fine, condition-coarse$ and $coarse-fine$). And then, the relation matching module selects out the positive predicted relations and discards all the other relations as results.

The Workflow is as follows.

- Pre-proces: When the program starts to process the context and the question. We first process the raw context and question. The preprocess splits the whole context and question into a bunch of single words, and substitutes special characters like space and line breakers with prescribed tokens.

- Sequence labeling: the first module labels every single word in the context and question with four labels, which are $None, Condition, Fine$ and $Coarse$.

- Labeling result process: After sequence labeling, we process the labeling results, restoring the digitized labels into their original phrase or sentence with location indices.

- Relation matching: the second module takes the restored and labeled phrases or sentences as input, forms all the potential relations as requested, and selects out the positive predicted relations as answers.

Solutions to challenges. In Section 3.2, we discuss the major challenges in developing a better performance BERT-HCMQA. We address these challenges through the methods as below.

To address the first challenge of having a more comprehensive understanding of context and distinguishing terminologies, we resort to the strength of BERT. As mentioned in 2.2, BERT, a pre-training model fine-tuned on a specific downstream task, has a deeper sense of context, and it's an excellent solution to extractive QA.

To address the second challenge of extractive multi-span question answering, we take advantage of token classification. Instead of considering this problem as traditional QA, we would rather regard it as a sequence label problem. We substitute the labels of the NER model, and fine-tune the model on our particular datasets. This method cleverly extracts multiple answers simultaneously by labeling.

To address the third challenge, which is to discover the hierarchical relationships between conditions, fine answers, and coarse answers. We take advantage of text classification. We combine context and two pieces of answers as input, and let the text classification model do a regression task, which figures out whether there is a specified relationship between the two answers in such a context.

4.2 Sequence Labeling

The sequence labeling module is based on the BERT token classification task. The BERT token classification task is commonly used in the Named-entity recognition (NER) task. (Jia et al., 2020). Although finding out conditions and coarse or fine answers seems obviously different from the NER task and is not related to labeling, we can replace the NEW labels and fine-tune the model on our own customized dataset. This labeling method helps us find the multiple answers simultaneously.

The sequence labeling module takes a bunch of single but ordered words as input, instead of a whole context.

The reason why we need to do such extra work is that we need to replace some special characters, like space or line breaker with special symbols ([PAD] for space and [unused1] for line breaker). Because these special characters will be discarded by BERT transformer. An input example is shown in Figure 2


```
{"tokens": ["女人肾阴虚吃什么好", "[SEP]", "女", "性", "肾", "阴", "虚", "在", "饮", "食",
"上", "建", "议", "多", "吃", "一", "些", "滋", "阴", "的", "食", "物", "，", "比", "如",
"山", "药", "、", "枸", "杞", "、", "石", "斛", "、", "玉", "竹", "、", "鸭", "肉", "、",
"莲", "子", "、", "百", "合", "等", "等", "，", "[unused1]", "[PAD]", "另", "外", "还",
"可", "以", "多", "吃", "一", "些", "黑", "色", "的", "食", "物", "，", "因", "为", "中",
"医", "认", "为", "五", "色", "入", "五", "脏", "，", "而", "黑", "色", "的", "食", "物",
"是", "入", "肾", "脏", "的", "，", "有", "很", "好", "的", "补", "肾", "功", "效", "。",
"[unused1]", "[PAD]", "必", "要", "的", "时", "候", "可", "以", "选", "择", "一", "些",
"滋", "阴", "补", "肾", "的", "中", "成", "药", "物", "如", "六", "味", "地", "黄", "丸",
"、", "大", "补", "阴", "丸", "、", "左", "归", "丸", "、", "知", "柏", "地", "黄", "丸",
"，", "归", "芍", "地", "黄", "丸", "等", "等", "，", "[unused1]", "[PAD]", "需", "要",
"在", "中", "医", "师", "的", "指", "导", "下", "辨", "证", "论", "治", "之", "后", "选",
"择", "合", "适", "的", "药", "物", "，", "患", "者", "一", "定", "要", "注", "意", "三",
"餐", "定", "时", "定", "量", "，", "不", "要", "暴", "饮", "暴", "食", "，", "[unused1]",
"[PAD]", "并", "且", "饮", "食", "以", "清", "淡", "、", "营", "养", "、", "易", "消", "
化", "为", "主", "。", "[unused1]"], "tags": [0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 2, 2, 2, 2, 2, 0, 0, 0, 3, 3, 0, 3, 3, 0, 3, 3, 0, 3, 3, 0, 3, 3, 0, 3, 3, 0,
3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 3, 3, 3, 3, 3, 0, 3, 3, 3,
3, 0, 3, 3, 3, 0, 3, 3, 3, 3, 3, 0, 3, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]}
```


Figure 2: Sequence Labeling module's input form

4.3 Relation matching

The relation matching module is based on the BERT test classification task. BERT text classification is widely used in sentiment analysis, which is a kind of analysis of regression (Sun et al., 2019). In our task, this task is fine-tuned to decide whether there is a specified relation between two given answers in the context.

Basically, the relation match module must take context and both of the two answers as input. Because the prediction about the relationship is dependent on understanding the context, Furthermore, in order to make the input more concise and informative. We trim the whole context into a piece which entails both the two answers. Besides, we take four special symbols $[head1]$, $[tail1]$, $[head2]$, $[tail2]$ to mark locations of two answers, which is illustrated in Figure 3

This form makes the input not only shorter but also more detailed, which makes the second module work much better. In addition, two relation matching models are used to separately select out the positive condition-fine/coarse relations and fine-coarse relations. Because the three different types belong to two more general types. This fine-grained sort of model helps us get more accurate predictions.

[CLS][head1]重度患者采取系统用药[tail1]，不仅口服[head2]维a酸类药物[tail2]，还会联合激素的药物进行治疗。[SEP]

Figure 3: Relation matching module's input form

5 Evaluation

5.1 Experimental Setup

In this part, we illustrate the evaluated methods, platforms and datasets.

Evaluated methods. Because we are expected to solve a problem in the Biendata CCKS2022 competition and try our best to get a good rank, there aren't any baselines or benchmarks. We constantly improve our model, guided by some general indicators, such as accuracy, precision, recall and so on.

Table 1: Performance of Two Modules

Module	Accuracy	Precision	Recall
Sequence Label	92.23%	75.67%	79.75%
Relation Match	99.75%		

Platform for BERT-HCMQA. The experimental platform for BERT-HCMQA is a server equipped with a 20-core/1-thread Intel(R) Core(TM) i7-12700K CPU at 3.6GHz and an NVIDIA RTX A6000 GPU. This graphics card has 10,752 GPU cores. Its theoretical maximum floating-point performance is 38.7 TFLOPS (tera floating-point operations per second). The GPU integrates 48GB of GDDR6, and the memory bandwidth can reach 768 GB/s. The operating system we use is Ubuntu 20.04.2 LTS. The torch version for BERT-HCMQA is 1.11.0. The CUDA Toolkit version for BERT-HCMQA is 11.3.

Datasets. There are no particular datasets used in our experiment. We only use the customized raw data provided by the CCKS2022 competition. We split the labeled data into two parts, where 80% is training dataset while 20% is validation dataset.

5.2 Results

In this part, we show the performance results of BERT-HCMQA.

Performance.The accuracy and some other indicators of two modules of BERT-HCMQA on customized datasets are shown in Table 1. The overall performance of our model ranks 6th in the competition up to now.

5.3 Discussion

Bottleneck. Although two modules get nice accuracy, the overall performance of our model still needs to be improved. We've tried our best and come up with several optimization methods, but still fail to further improve our model. After detailed analysis of our model, we believe that the bottleneck of our model lies in the first module. Because the first module has relatively low precision and recall. And when we get better results in the first module, the overall performance improves significantly.

How to improve. As described in the last paragraph, the bottleneck lies in the first part. We've tried some optimizations and tricks, but the improvement of the methods is trivial. We believe that if we want to further improve our model, we need to come up with a brand new method or model structure to get better results in the first part.

In general, we propose a new model based on BERT that performs well in extractive multi-span QA with conditions and hierarchy. Despite some shortcomings, our model is an effective attempt to solve the problem and has some referential value.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Philipp Cimiano, Christina Unger, and John McCrae. 2014. Ontology-based interpretation of natural language. *Synthesis Lectures on Human Language Technologies*, 7(2):1–178.

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

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, et al. 2017. Dureader: a chinese machine reading comprehension dataset from real-world applications. *arXiv preprint arXiv:1711.05073*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Chen Jia, Yuefeng Shi, Qinrong Yang, and Yue Zhang. 2020. Entity enhanced bert pre-training for chinese ner. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6384–6396.

Daniel D Lee, P Pham, Y Largman, and A Ng. 2009. Advances in neural information processing systems 22. Technical report, Tech. Rep., Tech. Rep.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. 2021. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Peng Xu, Davis Liang, Zhiheng Huang, and Bing Xiang. 2021. Attention-guided generative models for extractive question answering. *arXiv preprint arXiv:2110.06393*.

Wonjin Yoon, Richard Jackson, Jaewoo Kang, and Aron Lagerberg. 2021. Sequence tagging for biomedical extractive question answering. *arXiv preprint arXiv:2104.07535*.