UNDERSTANDING READING COMPREHENSION AND QUESTION ANSWERING USING CLINICAL REPORTS

A PREPRINT

Changye Li
Institute of Health Informatics
University of Minnesota
Minneapolis, MN 55455
lixx3013@umn.edu

Xinpeng Shen
Institute of Health Informatics
University of Minnesota
Minneapolis, MN 55455
shenx582@umn.edu

Ruyuan Wan
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
wanxx199@umn.edu

Jiaqi Zhou
School of Statistics
University of Minnesota
Minneapolis, MN 55455
zhou1186@umn.edu

October 23, 2020

Keywords Natural Language Processing · Question Answering · Machine Reading Comprehension

Introduction

Machine Reading Comprehension (MRC) and Question Answering (QA) are Natural Language Processing (NLP) tasks that train machines to answer a list of questions with the input context. In recent years, many researchers have been working on explorations and innovations in MRC and QA tasks using the open-domain dataset using the benchmark dataset SQuAD [12][1]. However, such tasks receive less attention in the biomedical area compared to their open-domain fellows.

A typical MRC and QA task application in the biomedical is clinical decision support, where a huge amount of text, including clinical notes and biomedical literature, then answer the questions from clinicians and physicians. There are several dataset, QA challenges and systems targeted at such services. For example, [9] provide a manually semantic annotated dataset on the biomedical field. Compared to previous studies, they annotate questions and answers that are related to health from consumers, not healthcare professionals. They annotate 2614 consumer health questions with several semantic layers: named entities, question topics, question triggers, and question frames. 67% of the questions come from consumer health emails received by the U.S. National Library of Medicine (NLM) with relatively long questions. The rest of the questions are from search query logs of MedlinePlus, a consumer-oriented NLM website for health information, which is relatively shorter[2]. [7] provide another annotated biomedical QA dataset: PubMedQA. PubMedQA has 1k expert-annotated, 61.2k unlabeled and 211.3 artificially generated QA instances. Compared to NLM's dataset, PubMedQA can also serve as QA system that works with reading comprehension style. Also, the

[1]For more details, please check https://rajpurkar.github.io/SQuAD-explorer/
[2]The dataset is publicly available at https://bionlp.nlm.nih.gov/CHIQAcollections/CHQA-Corpus-1.0.zip

answer for this dataset is either yes, no, maybe. However, MLM's answer is composed with several sentences or a paragraph[3].

BioASQ [15] is a another closed-domain QA benchmark challenge that focuses on biomedical area. The challenge have several sub-task: 1) classify large-scale biomedical documents onto ontology concepts to automate semantic indexing; 2) classify biomedical questions on the same concepts; 3) integrate relevant document snippets, database records, and information from the knowledge base; 4) deliver the retrieved information at the user-understandtandable level. In 2017, [1] organized the first consumer health question answering task in the TREC LiveQA task. This task focuses on automatically retreving answers to medical questions received by the U.S. National Library of Medicine (NLM). They provide a test dataset of 104 NLM questions for evaluation and 634 question-answer pairs for training. All training dataset for this task is publicly available[4]. The best participating system in this task achieves an average score of 0.637 on the scale mentioned in earlier section of this paragraph.

The CMU-LiveMedOAQA system [19] is yet another QA system serving in medical area. With given input, CMU-LiveMedOAQA first recognizes medical entity and attributes, then retrieves contents of the associated content, which involved with knowledge graph, finally ranks contents and generates the final answer. In the first stage, each token in the sentence is processed with `NLTK` package and ranked by TF-IDF scores from a background MedlinePuls Health Topics corpus. As results, they reduce the number of question types into 10: Treatment, Information, Susceptibility, Prognosis, Symptom, Diagnosis, Cause, Organization, Drug Information, and Drug Interaction. Predicting the question type then can transfer as multiple label classification problems. Therefore they utilize CNN model with Word2Vec word-embedding to solve the task. By building a hierarchy structure for medical entities with expertise, the system can crawl relevant information and find correct answers. The CMU-LiveMedOAQA system receives an average score of 0.356 on a 3 point scale and successfully answers 103 of 104 questions in the challenge. However, the error analysis shows that the small dataset to feed CNN could lead to over-fitting. Therefore, the wrong answers extracted by the knowledge tree model lowers the performance of the system. Also, their assumption of one question with one type only causes problem: the error analysis shows that some questions have multiple types. This system is a sub-system of CMU-OAQA system [17] that specifically designed for medical domain question answering. However, the sub-system does not beat CMU-OAQA in such challenge, whereas the parent system is built upon open-domain dataset.

In this paper, we use CliCR [14] to complete the MRC and QA task. CliCR constructs queries, answers and supporting context from 2005-2016 BMJ Case Reports, which is the largest online repository of detailed case report of rare diseases, unusual presentation of common conditions and novel treatment methods. Each case report contains a *Learning Points* section, which summarizes the key information from the reports. Those learning points are typically short paraphrased sentences from the case report and the queries are built upon them. The CliCR dataset contains 104,919 queries on 11,846 case reports and the answers for queries are single or multi-word medical entities. For each case report, [14] remove the HTML boierplate from the crawled reports, segment and tokenize the texts and annotate the medical entities using CLAMP [13]. The queries are created by replacing the medical entity in one learning point with a blank. For example, in a report of ADHD, the query "Patients with ADHD have higher incidence of __", where the missing entity "enuresis" is taken as the correct answer. Given the natural of clinical and biomedical text, the correct answer may have different string representation. Therefore, All different string representations under Concept Unique Identifier **CUI0014349**, including ENURESIS, Enuresis, are also considered as the correct answer for this query. The MRC task for CliCR can be represented as a tuple (q, p, A), where q is the query, built from the learning points; the context p is the full case report excluding learning points section; and A is the set of ground truth entities. Also, due to limited computing resources, we only select the first 20 case reports and their associated question and answer pairs for this work. Our experiments showed that BERT-whole-word-masking word-embedding and masked language model had the best performance and outperformed the gated-attention reader from CliCR original paper.

[3]The dataset is publicly available at `https://pubmedqa.github.io`

[4]`https://github.com/abachaa/LiveQA_MedicalTask_TREC2017`

In the following sections, we will discuss the preprocessing methods we used for model building, and models we applied as baseline models. We also provide a short error analysis with discussion at the end of our report. The code for this project is publicly available on GitHub[5].

Methods

Preprossessing

We implement the following methods for preprocessing:

- Remove the entity markers, e.g., `BEG__` and `__END`
- Remove newline markers, e.g., `\n`
- Remove non-ASCII characters
- Lower all tokens

Word-Embeddings

We use the following methods for word-embeddings in our models:

- NNLM text embedding - from homework 3
- BERT [6] pre-trained word-embedding from transformers [18] Python package
  - BERT-large-uncased-whole-word-masking
  - BERT-base-uncased
- Glove [11] pre-trained word-embedding
- BioWordVec [21], trained based on fastText [4] with PubMed and MIMIC-III data
- BioSentVec [5], trained based on Sent2Vec [10] with PubMed and MIMIC-III data
- Cui2Vec [2], trained based on Word2Vec with medical concepts with clinical notes and biomedical journals

Models

QANet

Similar as the generic encoder-decoder architecture for question answering tasks, the QANet model network can be separated into roughly 3 sections: embedding, encoder and attention. For the embedding layer, it used pre-trained Glove embedding to convert text into vector representations. Then the encoder layer consists of a positional encoding, depth-wise separable 1 dimensional convolution, self-attention and feed-forward structure with layer norm in between. Finally, attention layer is the core building block of the network where the fusion between question and passage occurs. The QANet uses trilinear attention function which was added to the Context-to-Query attention and Query-to-Context attention.[20]
Because we know that the answer is somewhere inside the passage, for each entity in the passage we just need to calculate the probability of the entity being the answer or not. If the predicted answer matches with a ground truth answer in the answer list, then we count it as a matched answer.

[5] `https://github.com/palooney/rc-clicr`

NNLM

We implemented the pre-trained embedding model NNLM [6] that was trained using the neural network probabilities language model framework. The embedding encode is learning a distributed representation of word. NNLM is also known as a sentence encoder. It first transforms each token into a 128 dimensional vector before combining them into a single vector. [3].
Here, we list the detail steps of the implementation:

- Extract concepts from the context: In this implementation, we assumed that the answer to the query is a concept that is contained in the given context. After extracting concepts, we can ignore all the remaining tokens.
- Apply embedding NNLM to the concepts list and the query: After we extracted the concept list, we applied the NNLM embedding model to the concept list and the query.
- Calculate the correlation: Calculating the correlation between the concepts vectors with the query vector. This step measure the distance between the embedding query and concept list.
- Pick the concept: We select the concepts with the highest Pearson correlation with the query, and use it as the answer to the query.

BERT-Based Methods

Before applying the BERT Masked language model, we need to preprocess the input context. Take the following case report as an example:

Query ▷ @placeholder from BEG__ amniotic band disruption__ END is a possibility .

Context BEG__ Isolated cranial distortion mimicking caput succedenum__ END from BEG__ amniotic band disruption__ END without any BEG__ neurological abnormality__ END Summary This report describes a term newborn with BEG__ isolated distortion in the left parietal bone__ END ....

Answer Isolated calvarial deformity mimicking caput succedenum

BERT whole word tokenized answer ['isolated', 'cal', '##var', '##ial', 'def', '##or', '##mity', 'mimic', '##king', 'cap', '##ut', 'su', '##cc', '##ede', '##num']

We replace ▷ *@placeholder* with mask tokens and concatenate the query and report with sentence separator [SEP]. We also keep a copy of the index for those [MASK] tokens for prediction. If the concatenated string has less than 512 tokens, then we pad the length of the input context to 512, as BERT model requested. In other words, the query would be preprocessed as *[MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK][MASK] [MASK] [MASK] from amniotic band disruption is a possibility. [SEP] isolated cranial distortion mimicking caput succedenum from amniotic band disruption without any neurological abnormalit summary this ....* Then we apply the BERT masked language model to predict those masked tokens.

BERT model uses transformers [16] attention mechanism to dispense with recurrence and convolutions entirely. The main idea behind the attention mechanism is that each time the model generates a prediction about the output word, it only uses parts of an input where the model considers as "most relevant information". That's being said, for the attention mechanism, the model only pays attention to some of the input words. The transformer is a novel architecture that help to solve sequence-to-sequence tasks while handling long-range dependencies with ease, which "relies on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution" as stated in

[6] The pre-trained NNLM model is available at NNLM `https://tfhub.dev/google/collections/nnlm/1`

[16]. In a transformer architecture, the encoder block has 1 layer of multi-head attention followed by another layer of feed forward neural network, whereas the decoder has an extra masked multi-head attention layer as the first layer.

Based on the BERT architecture, we implemented two state-of-the-art pre-trained models: bert-large-uncased-whole-word-masking and variation of bert-based-uncased.

- **bert-large-uncased-whole-word-masking** 24 layers(transformer blocks) and 16 attention heads
- **bert-based-uncased variant** 12 layers(transformer blocks) and 12 attention heads

  In the previous bert-large model, we tokenize the true answer entity and predict each masked WordPiece token independently from the whole vocabulary. It is based on the that we know the exact tokenized length of the true answer. Here we try to treat the whole entity as one token and since we know the answer will be one of the an marked entity in context, we force the model to choose answer from the candidates entities only.

  Besides, the BERT model only allows 512 tokens thus we just keep the first 512 tokens after concatenating the query and report context. Here, we select the report sentences most similar to the query, hoping that it will provide more useful information. The similarity is calculated by the cosine similarity between two sentences using BioBERT sentence embedding.

Results

We use the Answer Match (AM) as metric for our model evaluation. It is defined as the percentage of queries whose predicted answer are exact match of any answer from the answers list. As mentioned in previous sections, we only use the first 20 case reports from the dataset to evaluate our models. In table 1, we can see that BERT with whole-word-masking outperforms the gated-attention reader from the original CliCR paper and it is 0.05% less than the human performance.

Table 1: Model Performance on the Test Dataset

Model	AM (%)
NNLM	25
BERT large with whole-word-masking	**34.5**
BERT base variation	10
QANet	20
Gated-attention reader	22
Human performance	35

We also tried BioWord2Vec, BioSent2Vec, but they were too huge to load to our local computing resources. Also, we found that Cui2Vec is too limited that only about 20% of CUIs in our dataset are covered in Cui2Vec embedding and Cui2Vec does not handle out-of-vocabulary situation. Given the nature of CliCR data, we also tried SpanBERT [8], a pre-trained method that is designed to better represent and predict spans of text. However, it does not support the masked language model. However, SpanBERT does not work on masked language model as we used in previous sections.

Discussion & Conclusion

In this paper we present several models to solve the MRC and QA tasks using CliCR dataset. Our experiments showed that BERT masked language model with bert-whole-word-masking word-embedding has best performance. Although we are unable to load BioWordVec and BioSentVec, we believe that domain-specific word-embedding can significantly reduce the out-of-vocabulary (OOV) problem. For example, we found a predicted answer, “marginal zonexaymymomal lymphoma” as the correct answer is “monocytoid b-cell lymphoma”, where the word “zonexaymymomal” does not

exist. We suspect that the model was trying to predict tokens like "marginal zone lymphoma". Unfortunately, none of us have biomedical background so we are unable to give a clear interpretation for this prediction. Therefore, we believe that evaluations from experts could solve this issue.

As we discussed in the previous sections, all answers from the same query have one unique CUI but have differences on string representations. Therefore, in many cases, a correct answer could be mistakenly classified as wrong due to the different string representations, which we believe that expert evaluations can also help. If CUIs information can be used for the medical concept interpretation in our dataset, it is to be hoped to help the question answering. However, we found that CUIs are hard to combine into the system of MRC and QA tasks due to OOV issues.

We found a lot of state-of-the-art (SOTA) models for MRC and QA tasks are trained and fitted for SQuAD dataset, where queries start with interrogative words like when/what/where/how and only have one answer. On the other hand, such structure does not fit CliCR data. Therefore, it is hard to adapt SOTA models on our project.

At last, we would like to mention that we only use the first 20 case reports from CliCR dataset, therefore we do not fine-tune the BERT model. Also, the [MASK] tokens used in the training do not appear during fine-tuning, which increases the difficulty to adapt BERT masked language model to fit the text setting we have for CliCR. As a result, our performance in table 1 could be bias. Future work is necessary to adapt gap-filling models for MRC and QA tasks on CliCR dataset.

Member Contributions

- Changye Li
  - Preprocessing
  - BERT masked language model
- Xinpeng Shen
  - NNLM
- Ruyuan Wan
  - QANet
- Jiaqi Zhou
  - BERT variation

References

[1] ABACHA, A. B., AND DEMNER-FUSHMAN, D. On the role of question summarization and information source restriction in consumer health question answering. *AMIA Summits on Translational Science Proceedings 2019* (2019), 117.

[2] BEAM, A. L., KOMPA, B., FRIED, I., PALMER, N. P., SHI, X., CAI, T., AND KOHANE, I. S. Clinical concept embeddings learned from massive sources of multimodal medical data. *arXiv preprint arXiv:1804.01486* (2018).

[3] BENGIO, Y., DUCHARME, R., VINCENT, P., AND JANVIN, C. A neural probabilistic language model. *J. Mach. Learn. Res. 3*, null (Mar. 2003), 1137–1155.

[4] BOJANOWSKI, P., GRAVE, E., JOULIN, A., AND MIKOLOV, T. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics 5* (2017), 135–146.

[5] CHEN, Q., PENG, Y., AND LU, Z. Biosentvec: creating sentence embeddings for biomedical texts. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)* (2019), IEEE, pp. 1–5.

[6] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]* (May 2019). arXiv: 1810.04805.

[7] JIN, Q., DHINGRA, B., LIU, Z., COHEN, W. W., AND LU, X. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146* (2019).

[8] JOSHI, M., CHEN, D., LIU, Y., WELD, D. S., ZETTLEMOYER, L., AND LEVY, O. SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529* (2019).

[9] KILICOGLU, H., ABACHA, A. B., MRABET, Y., SHOOSHAN, S. E., RODRIGUEZ, L., MASTERTON, K., AND DEMNER-FUSHMAN, D. Semantic annotation of consumer health questions. *BMC bioinformatics 19*, 1 (2018), 34.

[10] PAGLIARDINI, M., GUPTA, P., AND JAGGI, M. Unsupervised learning of sentence embeddings using compositional n-gram features. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (2018).

[11] PENNINGTON, J., SOCHER, R., AND MANNING, C. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543.

[12] RAJPURKAR, P., JIA, R., AND LIANG, P. Know what you don't know: Unanswerable questions for squad, 2018.

[13] SOYSAL, E., WANG, J., JIANG, M., WU, Y., PAKHOMOV, S., LIU, H., AND XU, H. Clamp–a toolkit for efficiently building customized clinical natural language processing pipelines. *Journal of the American Medical Informatics Association 25*, 3 (2018), 331–336.

[14] SUSTER, S., AND DAELEMANS, W. CliCR: a Dataset of Clinical Case Reports for Machine Reading Comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana, 2018), Association for Computational Linguistics, pp. 1551–1563.

[15] TSATSARONIS, G., SCHROEDER, M., PALIOURAS, G., ALMIRANTIS, Y., ANDROUTSOPOULOS, I., GAUSSIER, E., GALLINARI, P., ARTIERES, T., ALVERS, M. R., ZSCHUNKE, M., ET AL. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *2012 AAAI Fall Symposium Series* (2012).

[16] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., AND POLOSUKHIN, I. Attention is all you need, 2017.

[17] WANG, D., AND NYBERG, E. Cmu oaqa at trec 2015 liveqa: Discovering the right answer with clues. Tech. rep., Carnegie Mellon University Pittsburgh United States, 2015.

[18] WOLF, T., DEBUT, L., SANH, V., CHAUMOND, J., DELANGUE, C., MOI, A., CISTAC, P., RAULT, T., LOUF, R., FUNTOWICZ, M., AND BREW, J. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv abs/1910.03771* (2019).

[19] YANG, Y., YU, J., HU, Y., XU, X., AND NYBERG, E. CMU livemedqa at TREC 2017 liveqa: A consumer health question answering system. *CoRR abs/1711.05789* (2017).

[20] YU, A. W., DOHAN, D., LUONG, M.-T., ZHAO, R., CHEN, K., NOROUZI, M., AND LE, Q. V. Qanet: Combining local convolution with global self-attention for reading comprehension. ICLR.

[21] ZHANG, Y., CHEN, Q., YANG, Z., LIN, H., AND LU, Z. Biowordvec, improving biomedical word embeddings with subword information and mesh. *Scientific data 6*, 1 (2019), 1–9.