LasigeBioTM at MEDIQA 2019: Biomedical Question Answering using Bidirectional Transformers and Named Entity Recognition

Andre Lamurias* and Francisco M. Couto

LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal

Abstract

Biomedical Question Answering (QA) aims at providing automated answers to user questions, regarding a variety of biomedical topics. For example, these questions may ask for related to diseases, drugs, symptoms, or medical procedures. Automated biomedical QA systems could improve the retrieval of information necessary to answer these questions. The MEDIQA challenge consisted of three tasks concerning various aspects of biomedical QA. This challenge aimed at advancing approaches to Natural Language Inference (NLI) and Recognizing Question Entailment (RQE), which would then result in enhanced approaches to biomedical QA.

Our approach explored a common Transformer-based architecture that could be applied to each task. This approach shared the same pre-trained weights, but which were then fine-tuned for each task using the provided training data. Furthermore, we augmented the training data with external datasets and enriched the question and answer texts using MER, a named entity recognition tool. Our approach obtained high levels of accuracy, in particular on the NLI task, which classified pairs of text according to their relation. For the QA task, we obtained higher Spearman's rank correlation values using the entities recognized by MER.

1 Introduction

Question Answering (QA) is a text mining task for which several systems have been proposed (Hirschman and Gaizauskas, 2001). This task is particularly challenging in the biomedical domain since this is a complex subject as answers may not be as straightforward compared to other domains. However, clinical and health care information systems could benefit greatly from automated biomedical QA systems, which could improve the retrieval of information necessary to answer these questions.

To help progress on this topic, the MEDIQA challenge proposed three tasks in the biomedical domain(Ben Abacha et al., 2019):

1. Natural Language Inference (NLI) - classify the relation between two sentences as either entailment, neutral or contradiction;

2. Recognizing Question Entailment (RQE) - classify if two questions are entailed with each other or not;

3. Question Answering (QA) - classify which answers are correct for a given answer and rank them.

We applied the same approach to all three tasks since they all could be modelled as text classification tasks. The objectives of the tasks were to classify pairs of text: sentence-sentence (NLI), question-question (RQE), and question-answer (QA). For the NLI task, we had three possible labels for each pair (entailment, neutral or contradiction), while the RQE task was a binary classification. For the QA task, each pair should be given a reference score representing how well the question is answered, which ranged between 1 and 4.

QA is a complex task that involves various components, and can be approached in several ways. While real-world scenarios require the retrieval of correct answers from larger databases, the QA task of this challenge simplified this problem by providing up to 10 answers retrieved by the medical QA system CHiQA. This system also provided a ranking to each answer, however, we observed that this ranking did not follow the manual ranking in most cases. We also observed that the retrieved

*alamurias@lasige.di.fc.ul.pt

answers could consist of one or more sentences. While in some QA scenarios, systems are required to select the text span that contains the answer (Rajpurkar et al., 2016), in this case it was only requested to re-rank the retrieved answers and classify which ones were correct. Although specific ranking algorithms exist (Radev et al., 2000), due to the nature of the task and the fact that the other two tasks involved comparison of text, we decided to train a classifier that compared each question with a potential answer, i.e., we predicted how good a text is at answering a given question.

Our approach uses pre-trained weights as a starting point, to fine-tune deep learning models based on the Transformer architecture for each of the challenge tasks (Vaswani et al., 2017). We used the BioBERT weights, trained on PubMed abstracts and PMC full articles, as the type of text should be more similar to the challenge data than the standard BERT models, which were trained on Wikipedia and BookCorpus. Furthermore, we incorporated other datasets into the RQE and QA tasks, and enrich the training data with semantic information obtained using MER (Minimal Named-Entity Recognizer) (Couto and Lamurias, 2018), a high computing performance named entity recognition tool.

2 Related Work

Deep learning approaches have lead to state-of-the-art results in various text mining tasks. These approaches make use of intermediary representations of the data to then fine-tune the weights to different tasks. Various models have been proposed, and, recently, the most successful ones have been based around the Transformer architecture (Vaswani et al., 2017). An advantage of this type of models is that we can use pre-trained weights such as those provided by BERT (Devlin et al., 2018) as a starting point to train a model for a specific task. These weights are tuned on large corpora using the Transformer architecture and have been shown to be effective language models. Different models were made available by the authors, with two variations of the architecture, and whether the true case and accent markers of the tokens are taken into account.

Due to the effectiveness of the BERT architecture, it has been already adapted for other domains. Lee et al. (2019) presented a model specific to biomedical language, which was trained on a large-scale biomedical corpora: 200k PubMed abstracts, 270k PMC full texts, and a combination of these two. Although the BioBERT models use the same vocabulary as the BERT models, the same WordPiece tokenization is performed. This way, even if biomedical documents contain words that were not in the original vocabulary, the tokenizer will separate these words into frequent subwords, minimizing out-of-vocabulary issues and keeping compatibility with the original models. The authors tested these models on several biomedical text mining tasks, obtaining competitive performance when compared with other state-of-the-art models.

One of the most common text mining tasks is entity recognition. This task is important because it is often the first step to other tasks, such as entity linking and relation extraction. MER is a simple but efficient approach to entity recognition, which uses vocabularies that can be extracted from ontologies to identify and link entities. MER focuses on simplicity and flexibility to reduce the processing time and the time necessary to adapt to other domains and entity types.

3 Methodology

3.1 Data Preparation

We participated in the three tasks using the same approach by modeling each one as a text classification problem. We used the training data of each tasks as document pairs, where a document could be a sentence, paragraph, question or answer. The NLI and RQE data had obvious labels, while for the QA data we used the reference scores. However, to distinguish between correct answers with more detail, we also incorporated the manually assigned ranks to the answers with reference scores 3 and 4:

$$\text{FinalScore} = \text{ReferenceScore} + \frac{11 - \text{Rank}}{10}$$

As there are up to 10 possible answers to each question, the final score will range between 1 and 5.

We removed instances where each element of the pair contained the same text, which happened sometimes in the RQE training set. Furthermore, we performed named entity recognition using MER to identify several types of entities mention in both questions and answers. We used MER since it can provide reliable entity mention annotations at a reasonable speed. We appended the

textual labels of the terms recognized to the end of the document, as a list separated by whitespaces. Since MER matches ontology concepts, if the synonym of a concept was recognized, it was converted to its main label.

We recognized terms from the: Human Phenotype Ontology, Disease Ontology, Chemical Entities of Biological Interest (ChEBI) ontology and Gene Ontology. Our objective was to add to each text a list of the entities that could summarize that text. We chose those ontologies because the questions were about biomedical subjects, and therefore the ontologies chosen should reflect the main domains of the data. The ontologies that we used comprise a total of 350,233 terms.

We also explored additional sources of data to train the classifiers, for the RQE and QA tasks. Regarding the RQE task, we employed the NLI dataset since it also contained entailment relations. Even though these datasets were generated from different corpora and the NLI dataset and for different purposes, we considered that additional data could still improve the results. To this end, we transformed the NLI dataset so that all entailment relations were labeled as positive, and the neutral and contradiction as negative.

For the QA task, we added one of the suggested MedQuAD datasets, namely the CancerGov dataset. Although all these additional datasets had a similar structure, we did not have time to train and test which ones would be more helpful for this task. These datasets contained only examples of correct answers, which we assigned the reference score 4, since it could skew the trained classifier towards higher scores. To balance this, we generated incorrect answers from the other QA of the same document. We assumed that if an answer was correct for one question, it would be incorrect for the other questions about the same topic. To make sure this was true, we took into account the "qtype" parameter of each question, since it is unlikely that questions of different types would have the same answers. This parameter indicated the nature of the question in the context of the main topic of the document. For example, a document about a specific cancer type could have the following "qtypes": information, symptoms, exams and tests, outlook, and treatment.

Run	Training data	Dev	Test
1	NLI training set	0.836	0.724

Table 1: Accuracy obtained on the NLI task.

3.2 System architecture

We adapted the pytorch implementation of BERT[1]. As such, we used the WordPiece tokenization and Adam optimizer that are implemented by default. We used the BioBERT PubMed+PMC pretrained weights, which are based on the bert-base-cased model. The authors chose this model as many biomedical entities are case sensitive. We initially tested with the standard BERT weights, and observed an improvement when using the BioBERT weights instead. A model fine-tuned to the clinical domain, which is the domain of the documents of this challenge, would be more appropriate, but not such pre-trained model was available at the time.

Using the data previously described, we trained variations of the same model, focusing mostly on the RQE and QA tasks. These variations consisted of the additional datasets previously described, but also different training parameters, such as initial training rate, number of epochs, batch size and maximum sequence length. We started with the default values and made incremental changes to understand if we could improve the results on the validation set, while training just with the provided training set. After setting the best parameters, we then trained the classifiers on the additional datasets.

For the NLI, we tested only the baseline approach, which consisted in using the BioBERT weights fine-tuned for the task.

4 Results and discussion

We submitted one run to the NLI task, three runs to the RQE task and four runs to the QA task. We focused mainly on studying the effect of different training data on the performance of the classifiers.

We evaluated on the development sets that were provided for each task, and then submitted our predictions for the test sets. The scores obtained for the development and test sets of each task are shown in tables 1, 2 and 3, as well as the differences between each run.

[1] https://github.com/huggingface/pytorch-pretrained-BERT

Run	Training data	Dev	Test
1	RQE training set	0.732	0.481
2	RQE training set + NER	0.**752**	0.481
3	RQE and NLI training set	0.749	**0.485**

Table 2: Accuracy obtained on the RQE task.

We can see that the accuracy obtained during the development phase was considerably higher than on the test set. This could have been due to the test set containing other type of questions from the development set, or due to over-fitting of the hyper-parameters on the development set, which limited the performance of the model. Both on the test and development set, we obtained high accuracy on the NLI task, for which we submitted only one run. The NLI data was generated by asking experts to give one example of each class (neutral, entailment and contradiction) to a series of statements. As such, this dataset is highly regular and the model was able to learn from it.

On the development set, we can see that adding the named entities recognized by MER (Run 2 of RQE and QA) improved the accuracy. However, this effect did not occur on the test set; for the RQE task, it did not change the accuracy and it decreased the accuracy of the QA task. On the other hand, adding external training data (Run 3) had a positive effect on the test set results of both tasks, improving the accuracy of the QA task.

For the QA, we also trained a classifier using both the training and development datasets (Run 4). We could not evaluate this classifier on the development set since it had already seen those examples and the results would have been biased. However, this classifier achieved the best test set accuracy and Mean Reciprocal Rank (MRR) of the four runs submitted to this task.

The best results obtained with our approach were on the NLI task. However, we considered the QA task to be the main task of the challenge and put most effort into it in terms of exploration hyper-parameter tuning. Since the organizers considered the accuracy to be the main metrics, we optimized our system to that metric. While the MRR was high on all three runs, the Spearman's coefficient was generally much lower. This means that although our system was able to detect correct answers to a certain degree, their ranking matched poorly with the gold standard.

5 Conclusions and Future Work

For the MEDIQA challenge, we developed a system that could be used for the 3 proposed tasks with minimal changes. This was possible due to the recently introduced Transformer architecture, along with pre-trained weights that severely reduce the training time necessary to generate a language representation model. The training data provided for each task was used to train classification models for each task. We also explored external datasets to improve the models of the RQE and QA tasks. We observed that adding more data to train the model leads to better results on the test set, as expected.

In the future we will improve the capacity of the models to classify new data by adding more external training data. We observed that Runs 3 and 4 of the QA task achieved higher scores, which could have been due to the larger training set employed to train the models. While for Run 3 we used only one additional set, there were 9 more available of the same type, which were not used due to time constraints. A similar strategy could be used to find more pairs of questions with an entailment relation.

Another way to enrich the training set would be to automatically retrieve the descriptions of the entities identified in the text, or their ancestors, as they also provide useful information about entities. A similar approach was shown to improve the results of a relation extraction task using deep learning (Lamurias et al., 2019).

References

Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. 2019. Overview of the mediqa 2019 shared task on textual inference, question entailment and question answering. In *Proceedings of the BioNLP 2019 workshop, Florence, Italy, August 1, 2019*. Association for Computational Linguistics.

F. Couto and A. Lamurias. 2018. MER: a shell script and annotation server for minimal named entity recognition and linking. *Journal of Cheminformatics*, 10(58).

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*.

Lynette Hirschman and Robert Gaizauskas. 2001. Natural language question answering: the view from here. *natural language engineering*, 7(4):275–300.

		Dev			Test		
Run	Training data	Accuracy	Spearman	MRR	Accuracy	Spearman	MRR
1	QA training set	0.782	0.067	0.760	0.585	**0.220**	0.843
2	QA training sets + NER	**0.791**	**0.198**	0.840	0.551	0.026	0.733
3	QA training sets + CancerGov	0.756	0.183	**0.920**	0.600	0.201	0.870
4	QA training and dev sets	-	-	-	**0.637**	0.211	**0.910**

Table 3: Results obtained on the QA task. Spearman: Spearman's Rank Correlation Coefficient; MRR: Mean Reciprocal Rank.

A. Lamurias, Diana Sousa, L. Clarke, and F. Couto. 2019. BO-LSTM: classifying relations via long short-term memory networks along biomedical ontologies. *BMC Bioinformatics*, 20(10).

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*.

Dragomir R Radev, John Prager, and Valerie Samn. 2000. Ranking suspected answers to natural language questions using predictive annotation. In *Proceedings of the sixth conference on Applied natural language processing*, pages 150–157. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.