Human Language Engineering

Final Project: MEDIQA19 NLP Challenge

Master in Artificial Intelligence - Universitat Politècnica de Catalunya

Gonzalo Recio Domènech (gonzalo.recio@upc.edu)
Jana Reventós Presmenes (jana.reventos@estudiantat.upc.edu)

10th of January 2021

Abstract: This work presents an approach to solve a Question Answering shared task from MEDIQA19 NLP challenge. We present a comparison between several pre-trained models on the medical domain and different architectures. Despite our limitations on computational resources, our method manages to beat the provided baseline and achieves results that are among the top-10 teams in the challenge. The code related to this work can be found in this repository.

1 Introduction

As users struggle to navigate the wealth of on-line information now available, with the massive collection of full-text documents the need for automated question answering systems is urgent. We need systems that can display a correct answer with sufficient context given a specific question.

Question Answering (QA) systems address this problem. These engines are used to answer questions in the form of natural language, are dialogue-based systems, and have a wide range of applications. Typical applications include intelligence voice interaction (e.g. Siri, Alexa, Google assistant...), online customer service, knowledge acquisition, and more.

In QA tasks the neural model receives a question regarding a specific context and it's required to retrieve the most appropriate answer. In this work, we address the problem of Question Answering in the medical domain. The main objective of our project is to solve the third task of the MEDIQA 2019 challenge which consists of filter and re-rank the answers provided in the ChiQA artificial intelligence system. The MEDIQA 2019 competition aims to develop relevant methods, techniques, and gold standards in the medical domain for inference and entailment to improve domain-specific Information Retrieval and Question Answering systems.

To solve this task we design two systems built upon the BERT [Devlin et al., 2018] and ELMo [Peters et al., 2018] models which are trained in large domain corpora. Nonetheless, since the MEDIQA 2019 challenge is set in the medical domain, we compare different variations of BERT and ELMo models: BioBERT [Lee et al., 2019], Bio ClinicalBERT [Huang et al., 2020], SciBERT [Beltagy et al., 2019], and BioELMo [Jin et al., 2019].

2 What is Question Answering?

Question Answering can be viewed as an extension of Information Retrieval (IR) frameworks. In IR the user information needs are expressed through a query, and it is used for retrieving from a dataset the best set of documents that satisfy the user information needs. The main objective of QA systems

is to get answers to questions rather than the full document. In QA systems, the user query consists of a natural language question, and the output is a valid and accurate answer to the user question asked.

The job of producing answers to questions is associated with the type of questions asked. The choice of the domain as a basis of the classification of QA is crucial. Some users need general information on a general topic while other users need specific information from a particular application domain. QA systems can be classified into two types:

- Open-domain QA systems: are not restricted to any specific domain and provide a short answer to a question.
- Closed-domain QA systems: there is a restriction of the domain and questions are related to a specific domain.

To solve the third task of the MEDIQA 2019 challenge we design a closed-domain QA system as we are dealing with questions in the medical field.

2.1 Brief history of QA

The origin of Question Answering systems is found in the '60s of the last century. BASEBALL [Green Jr et al., 1961] was one of the early question answering systems that answered questions about the US baseball league. In 1964 Simmons et al. [Simmons et al., 1964] did the first exploration of answering questions from an expository text based on matching dependency parses of a question and answer. Later on, in 1993 Kupiec developed the MURAX approach [Kupiec, 1993] for answer questions over an online encyclopedia using an IR system and shallow linguistic processing.

Research in QA began to focus on open-domain questions when the Text REtrieval Conference (TREC) started a QA track in 1999. TREC is a series of workshops focusing on rigorously investigate answering fact questions over a large collection of documents. In 2003 QA system challenges acquired a multilingual dimension in the CLEF QA track. Currently, there are some other famous competitions, such as the SQuAD leader-board and MS MACRO leader-board.

In February of 2011, a computer called Watson[1] competed in Jeopardy! TV quiz show against the biggest all-time champions. Watson is a computer running software developed by IBM Research, is a question-answer computer system capable of answering questions in natural language. The system is trained primarily by data as opposed to rules, and it is described as a heterogeneous ensemble of experts, each part is specialized in solving a specific sup-problem.

Nowadays, famous systems that are accessible through the web are START (MIT), ASK.com, Wolfram Alpha and many more.

2.2 QA in the medical domain

Thanks to the recent developments in the field of biomedicine have to lead to the rapid growth of biomedical literature available online. Generic search engines seldom have any advantages in the medical domain because the emphasis of the system is more on keywords rather than phrases, as desired. They are usually trained with so-called "wh-words" named factoid type questions (what, which, when, who, how) and it is not appropriate for the medical domain for the following reasons [Andrenucci, 2008]:

[1] https://www.ibm.com/ibm/history/ibm100/us/en/icons/watson/

1. Questions about patients' care deal with diagnosis, treatments, symptoms, prognosis, outcome, or information of diseases. It requires a system capable of analyzing question types that are different from the factoid type questions.

2. Answers to "when" questions in medical are usually related to relative time rather than absolute time/dates (*QA: "When should I eat my medicine?" A: "One hour before lunch"*). This fact requires a deeply semantic representation of the user question.

For these reasons, in medical applications, correct or incorrect answers can be retrieved if contextual information is not understood, since the context can provide more evidence. Following this line, [Ben Abacha and Demner-Fushman, 2019] proposed a novel QA approach based on Recognizing Question Entailment, with the introduction of the MedQuAD medical question-answer collection, and showed empirical evidence supporting question entailment for QA.

Recent progress of biomedical text mining models was made possible by the advancement of deep learning techniques used in Natural Language Processing (NLP). For instance, Long Short-Term Memory (LSTM) and Transformer [Vaswani et al., 2017] models have greatly improved the performance in Name Entity Recognition (NER), Relation Extraction (RE) IR, and QA tasks over the last years.

Nonetheless, applying state-of-the-art (SoTA) NLP models in biomedical text mining has limitations as recent word representation models such as ELMo [Peters et al., 2018] and BERT [Devlin et al., 2018] are trained and tested on datasets containing general domain texts (e.g. Wikipedia). Also, the word distribution in general and medical domain corpora are different, which makes it difficult to use these kinds of models.

As a result, research in biomedical text mining models have been done during the last years to adapt the versions of word representations: BioBERT [Lee et al., 2019], SciBERT [Beltagy et al., 2019], Bio ClinicalBERT [Huang et al., 2020], BioELMo, [Jin et al., 2019] among others.

In this work, we design a system to solve the third task of the MEDIQA 2019 challenge based on adapted versions of the BERT and ELMo word representation models. We develop a benchmark comparing the ChiQA system baseline and different pre-trained models to discover which is the best approach for our task.

3 Problem Definition: MEDIQA 2019

In this work, we address the MEDIQA 2019 shared task-organized at the ACL-BioNLP workshop. This challenge is motivated by a need of developing novel frameworks for inference and entailment in the medical field, and its applications such as specific information retrieval and question answering systems. The MEDIQA 2019 competition includes three NLP tasks:

- Task 1: Natural Language Inference (NLI) - Identify three inference relations between two sentences: entailment, neutral, and contradiction.
- Task 2: Recognizing Question Entailment (RQE) - Identify entailment between two questions in the context of QA. We say that a question A entails a question B if every answer to B is also a complete or partial answer to A.
- Task 3: Question Answering (QA) - Filter and improve the ranking of automatically retrieved answers from the CHiQA system.

After attending the HLE lessons we gained interest in developing a specific question-answering system in a challenging domain using SoTA models. With this motivation in mind, in this work, we are going to focus only on the 3rd task of the MEDIQA19 challenge, which is related to Question Answering.

3.1 Task 3: Question Answering

The objective of this task is to filter and improve the ranking of automatically retrieved answers from the CHiQA system. To provide access to high-quality health-related information online and support customer services, the U.S. National Library of Medicine (NLM) started the Consumer Health Information and Question Answering (CHiQA) project.

The CHiQA goal is to provide a question-answering engine able of providing reliable ranked answers from patient-oriented resources to health questions asked by consumers, Figure 1.

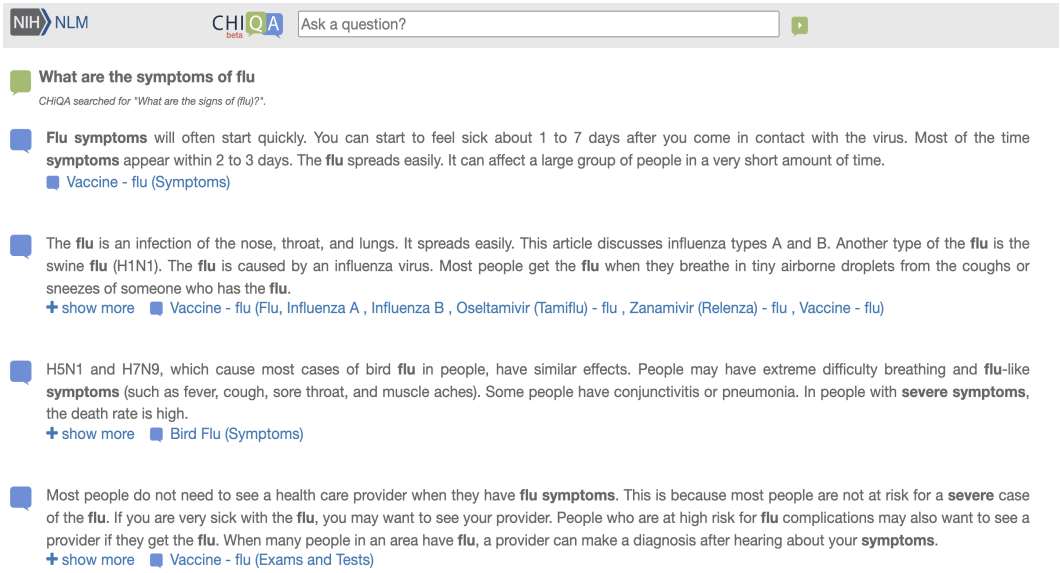


Figure 1: CHiQA system example.

The main problem of the actual CHiQA baseline system is that for a specific user question, the engine generates answers that often are not valid or accurate enough because:

- The system displays not entailed answers, in other words, answers that don't have any relation with the user question.
- The system doesn't rank the answers from more appropriate to less. For instance, the second answer should be the first in the ranking whereas the first answer should be ranked in the third position.

In this work, we want to improve the CHiQA ranking system to provide more suitable answers for questions asked by consumers. So, given a medical question and a set of candidate answers our implementation has to be capable of:

- Rank the answers from more appropriate to less.
- Classify answers as entailed to the question or not.

3.2 Datasets

The challenge provides data for the third task. Training, validation, and test sets were created by submitting medical questions to the CHiQA system. The MEDIQA-QA training data is constituted

with two datasets of medical questions and the associated lists of answers retrieved by the medical QA system CHiQA. These datasets are the following:

1. "MEDIQA2019-Task3-QA-TrainingSet1-LiveQAMed.xml": 104 consumer health questions covering different types of questions about diseases and drugs and 839 associated answers retrieved by CHiQA and manually rated and re-ranked.
2. "MEDIQA2019-Task3-QA-TrainingSet2-Alexa.xml": 104 simple questions about the most frequent diseases and 862 associated answers.

The MEDIQA-QA validation set consists of 25 consumer health questions and 234 associated answers returned by CHiQA. The MEDIQA-QA test set consists of 150 consumer health questions and 1.107 associated answers.

For the set of questions, each associated answer is provided with three different ranks:

- SystemRank (ranking from CHiQA system) : baseline rank
- Reference Rank (manually re-ranked answers from experts): correct rank
- ReferenceScore: is the manual judgment/rating of the answer [4: Excellent, 3: Correct but Incomplete, 2: Related, 1: Incorrect]. For the answer classification task: answers with scores 1 and 2 are considered as incorrect (label 0), and answers with scores 3 and 4 are considered as correct (label 1).

Listing 1: Task 3 QA dataset example

```
<QuestionText>What causes Flu?</QuestionText>
 <AnswerList>
    <Answer AID="2_Answer1" SystemRank="1" ReferenceRank="7"
       ReferenceScore="2">
        <AnswerURL>https://medlineplus.gov/ency/article/007444.htm</
           AnswerURL>
        <AnswerText>Your baby and the flu (Information): FLU SYMPTOMS
            IN INFANTS AND TODDLERS The flu is an infection of the
           nose, throat, and (sometimes) lungs....
        </AnswerText>
    </Answer>
    <Answer AID="2_Answer2" SystemRank="2" ReferenceRank="1"
       ReferenceScore="4">
        <AnswerURL>https://www.nlm.nih.gov/medlineplus/ency/article
           /000080.htm</AnswerURL>
        <AnswerText>What causes Flu?: The flu is caused by an
           influenza virus...
        </AnswerText>
    </Answer>
        ...
```

In addition, the challenge mentions that the Medical Question Answering Dataset (MedQuAD) of 47k question-answer pairs can be used to retrieve answered questions that are entailed/related from the original questions [Ben Abacha and Demner-Fushman, 2019]. We use this dataset in the training to provide additional information and develop a more consistent QA ranking system.

3.3 Evaluation

In this section we explain how models are evaluated for measuring their performance and which is the baseline that it is expected to outperform.

3.3.1 Evaluation metrics

The evaluation of our system uses the same metrics that the MEDIQA 2019 challenge requires in order to compare our results with the participants of the QA task. For the QA task the evaluation is based in four metrics:

- Accuracy: on the predicted labels whether the answers are entailed to the questions. Here the accuracy will be computed by the prediction of the incorrect answers (label 0).
- Precision: number of correct ranked answers divided by the total number of retrieved answers for an specific question.

$$Precision = \frac{tp}{tp + fp}$$

- Mean Reciprocal Rank (MRR): evaluates any process that produces a list of possible responses to a sample of queries, ordered by probability of correctness.

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i}$$

- Spearman's Rank Correlation Coefficient (Spearman's rho): Penalizes the differences (d) on predicted ranks and true ranks.

$$\rho = 1 - \frac{6 \sum d}{n(n^2 - 1)}$$

To compute all these metrics, the challenge organizers provided an evaluation script to automatically do that and avoid mistakes.

3.3.2 Baseline Systems

For this work, we use two baseline systems. The QA baseline system is the CHiQA question-answering engine [Demner-Fushman et al., 2019], which was used to provide the answers for the QA task. Besides, we use the BERT model as baselines to provide evidence of the benefits of using medically adapted versions of both approaches (BioBERT, Bio ClinicalBERT, SciBERT, and BioELMo).

4 State of the Art

A published paper about the results obtained in the shared task [Ben Abacha et al., 2019]. Table 1 shows the top 10 teams that achieved best results. The last row represents the baseline results achieved by the CHiQA system.

The question answering task was challenging as the maximum accuracy was 78%, but, most systems did well in the first retrieved answers with a best MRR score of 96.22%. We also note that Spearman's rho metric is the most unstable and does not seem to have a clear correlation with the model accuracy performance.

In general, teams used their NLI and/or RQE BERT models for the QA task. Even though in this work we don't address the first two tasks of the MEDIQA 2019 challenge, most of the participants used pre-trained variations of BERT. The reason for that is because contextual word embedding models such as ELMo and BERT have dramatically improved performance for many natural language processing tasks over the last years. However, these models have been minimally explored on specialty corpora, such as clinical text.

Rank	**Team**	**Accuracy**	**Precision**	**MRR**	**Spearman's rho**
1	DoubleTransfer	0.780	0.8191	0.9367	0.238
2	PANLP	0.777	0.7806	0.9378	0.180
3	Pentagon	0.765	0.7766	0.9622	0.338
4	DUT-BIM	0.745	0.7466	0.9061	0.106
4	DUT-NLP	0.745	0.7466	0.9061	0.106
6	IITP	0.717	0.7936	0.8611	0.024
7	lasigeBioTM	0.637	0.5975	0.91	0.211
8	ANU-CSIRO	0.584	0.5568	0.7843	0.122
9	Dr.Quad	0.565	0.6679	0.6069	0.009
10	ARS NITK	0.536	0.5596	0.6293	0.196
-	*Provided Answers*	*0.517*	*0.5167*	*0.895*	*0.315*

Table 1: Official results of the top-10 participants of the MEDIQA19 QA Task

The pre-trained BERT variations used by participants are explained below:

- SciBERT [Beltagy et al., 2019] variant of the original BERT trained with full text scientific articles (PubMed).
- BioBERT [Lee et al., 2019] is initialized with the original BERT model and then pretrained on biomedical articles from PMC full text articles and PubMed abstracts.
- Bio ClinicalBERT [Huang et al., 2020] is initialized with the original BERT model and then pre-trained on clinical notes from the MIMICIII dataset.
- Multi-task Deep Neural Networks for NLU by incorporating BERT as its shared text encoding layers [Liu et al., 2019] used for the three tasks. The architecture allows to solve the problem of answer ranking.

5 Question Answering approach

In this section, we introduce the approach that we use to solve the Question Answering task of the MEDIQA 2019 competition. There are three main challenging aspects of this task:

1. A **small** Q&A dataset with ranked answers is provided. To overcome this drawback we can use other datasets such as MedQuAD (47k questions and answers) and transfer learning technique.
2. We are working on a **specific** medical domain. Currently, BERT pre-trained model variations that have been trained on specific medical corpora exist.
3. The system should provide **ranked answers**. It is difficult to find which answer is better given 2 or more good answers. Some approaches that could be implemented to address this problem are to use binary classification between answers, train a similarity metric between QA encodings or use a probability maximization.

Due to the low computational resources for experimenting, we exploit transfer learning and we use the provided train, validation, and test dataset, which have limited samples, plus the additional MedQuAD datasets. We implemented two different QA models. The first is build upon three variations of pre-trained BERT models, BioBERT, SciBERT, and Bio ClinicalBERT, and the second model is built on a BioELMo pre-trained RNN model.

The idea is to work with sentence embeddings (mapping both the questions and the answers to an embedded space), and then train a similarity/entailment function between a given question and a given answer. Therefore, the model gets as an input a question and an answer and outputs the probability that the *answer* is a suitable response for the *question*. The overall idea of the architecture is illustrated in Figure 2.

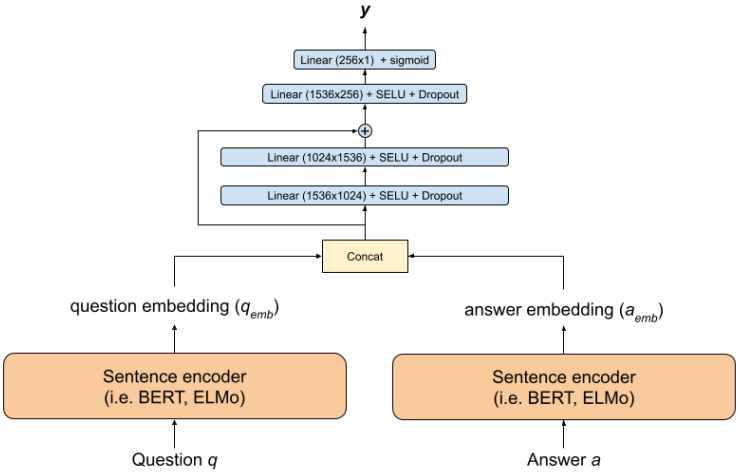


Figure 2: Our proposed model architecture.

Note that we added a residual connection in the last feed-forward network so that we allow the model to skip training for the layers that are not useful and do not add value in overall accuracy.

For both model implementations, we use two types of loss metrics to train our deep learning model: Binary Cross-Entropy Loss and Ranking Loss. The Binary Cross Entropy measures the error of the entailment prediction, and the Ranking Loss computes the ranking distance to the ground truth between answers associated with the same question.

5.1 Question-answer collection

Before describing the models we explain the methodology implemented to collect the QA data for the training set. We developed an engine that extracts for each training question the list of associated answers plus their system rank, reference rank, and reference scores (from 1 to 4).

After doing some experiments with just the MEDIQA QA Task 3 training datasets we realized that our system obtained limited performance as the ranking and answer classification was not still appropriate, additional biomedical and clinical information was needed. We used some external data from the MedQuAD dataset for training.

5.1.1 Additional question-answer collection

MedQuAD [Ben Abacha and Demner-Fushman, 2019] includes up to 47,457 medical question-answer pairs created from 12 NIH websites (e.g. cancer.gov, niddk.nih.gov, GARD, MedlinePlus Health Topics). The collection covers 37 question types (e.g. Treatment, Diagnosis, Side Effects) associated with diseases, drugs, and other medical entities such as tests.

In order to improve the system performance, we decided to extend the training set with the available MedQuAS corpus. For each question and question type, we take the answers associated with the question type with classification label 1 (the same as having a reference score of 3 or 4) and we call them positive examples. We create negative examples, with classification label 0 (the same as having a reference score of 1 or 2), by taking for each question and question type the answers associated with different question types. Also, we don't include the system rank because it doesn't exist, and we take as a reference rank, for each question, 1 for positive answers (meaning that any of those could be ranked as first) and large enough value for negative answers.

This methodology may improve the learning process of the models as we teach the system what answers do we want to retrieve for a specific question type and what answers we don't want to retrieve. This is crucial because often the system gives answers that are related to the question but are not answering the appropriate knowledge. For instance, sometimes if the question is *"What are the symptoms of flu?"* the provided answer describes *"What is flu."*, which demonstrates that the model doesn't take into consideration the whole question context, thus, a wrong answer is retrieved.

5.2 BERT-based model

BERT [Devlin et al., 2018] is a bidirectional transformer-based model [Vaswani et al., 2017] trained with a masked language modeling and a next sentence prediction task on a corpus of around 3.3B words. The masked language model masks some percentage of the input tokens randomly and tries to predict them, and the next sentence prediction task helps to understand the relationship between two sentences. Besides, BERT obtains SoTA performance on most NLP tasks, and it requires minimal architectural task-specific modification

The general BERT architecture incorporates information from bidirectional representations, rather than unidirectional representations. This framework fits in biomedical text mining as the relationship between biomedical entities is complex, and capturing the information in both directions is crucial in biomedical corpora.

However, BERT cannot take text longer than the maximum length (512 tokens) as input since the maximum length is predefined during pre-training. When BERT is applied to NLP tasks one of the main problems comes when inputs should be truncated by the maximum sequence length which decreases the performance since the model cannot capture long-term dependencies and global information across the whole document. Also, sequences need to be padded if they are shorter than the maximum length (with special token "`[PAD]`").

BERT is pre-trained on English Wikipedia and BooksCorpus datasets which are limited in biomedical domain texts. Biomedical corpora incorporate domain-specific proper nouns (i.e. influenza, COPD) and terms (i.e. hypoventilation, antimicrobial) that are not commonly used in general propose corpora [Lee et al., 2019]. As a result, using BERT in biomedical-specific NLP tasks does not give a relevant performance.

5.2.1 Pre-trained BERT variations

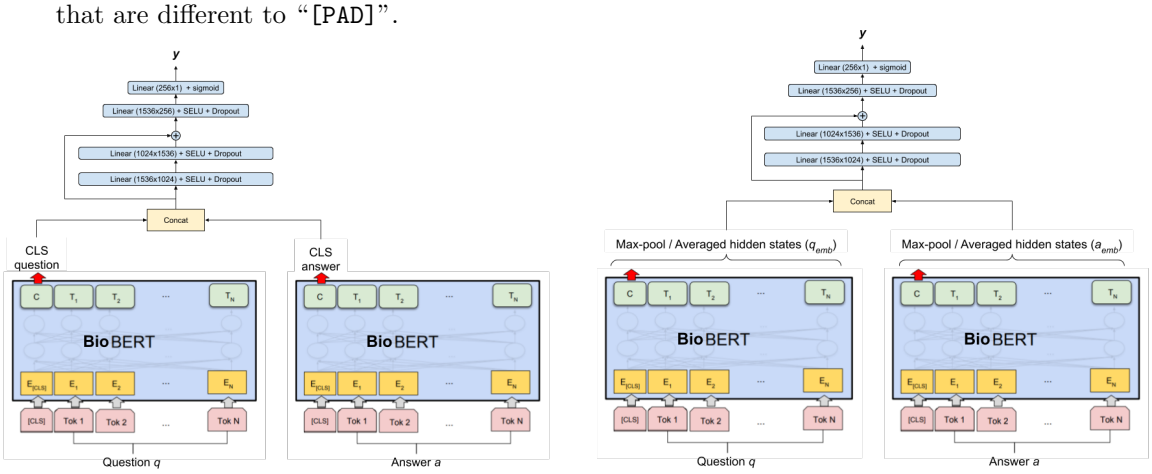
In this work, we use three BERT-based pre-trained language representation models on the biomedical and clinical domain, BioBERT, SciBERT, and Bio ClinicalBERT. BioBERT model [Lee et al., 2019] was initialized from BERT and trained in the 4.5B words PubMed abstracts and 13.5B words PubMed Central-Full text articles. SciBERT [Beltagy et al., 2019] follows the same architecture as BERT but was pre-trained on scientific texts from the SentencePiece library, the total corpus size was about 3.17B tokens. On the other hand, Bio ClinicalBERT model [Huang et al., 2020] was initialized from BioBERT and trained in the 2 million notes MIMIC-III v1.4, a database containing electronic health records from ICU patients at the Beth Israel Hospital in Boston, MA.

We fine-tune the BioBERT, SciBERT, and Bio ClinicalBERT models on the biomedical text mining MEDIQA-QA task. As sentence embedding vectors for questions and answers we considered two approaches:

- Use the CLS hidden token produced by the model (Figure 3a). The CLS is the first token of every sequence and acts as an "aggregate representation" for classification tasks. However, it is not the best choice for a high-quality sentence embedding vector as BERT does not generates meaningful sentence representations according to the BERT author Jacob Devlin.

  Nonetheless, as we are fine-tuning the model, the CLS token does become meaningful because the last hidden layer of this token is used as a sentence vector for sequence classification.

- Doing an average pooling or a max-pooling of the hidden layers of all the token embeddings produced by the model, obtaining a single sentence embedding vector (Figure 3b). In this case, for padded sequences we use an attention mask to only average/max-pool, the hidden tokens that are different to "`[PAD]`".



(a) Using CLS token.

(b) Using average of token hidden states.

Figure 3: Different architectures with BERT-based. BERT images retrieved from [Devlin et al., 2018]. The BERT modules can be changed by any variation pre-trained on medical corpora.

Question and answer sentence embeddings are concatenated and fed to the last linear layers of our model architecture, described in Figure 2.

5.3 ELMo-based model

ELMo [Peters et al., 2018] is a deep contextualized word representations model that, unlike other models such as BERT, takes the entire input sentence by using bidirectional recurrent and convo-

lutional layers. Contextualized word embeddings are representations that compute both the word meaning along the context information.

As general architecture ELMo consists of one forward and one backward language model, so, ELMO's hidden states have access to both the next word and the previous word. Each hidden layer is a bi-directional LSTM, in that way the model can view hidden states from either direction. T

Word representations are computed on top of two-layer biLMs with character convolutions, that compute a context-independent token representation (via token embeddings or a CNN over characters) that are then passed through layers of biLSTMs. This setup allows doing semi-supervised learning, where the biLM is pretrained at a large scale and easily incorporated into a wide range of existing neural NLP architectures.

Hence, ELMo allows us to build a sentence embedding of the whole sequence (without truncating) and is capable of capturing sentence context information in both directions. These two main characteristics are beneficial in biomedical text mining tasks, often medical questions and answers are longer than the maximum length permitted in BERT pre-trained models. Furthermore, this bi-directional architecture allows the model to learn complex relations between biomedical entities.

As BERT model, ELMo was pre-trained on a huge corpus of monolingual data from a general domain, so using directly ELMo in biomedical-specific NLP tasks does not give a good performance.

5.3.1 Pre-trained ELMo variation

The fourth pre-trained model that we use in this work is BioELMo [Jin et al., 2019], which is a domain-specific version of ELMo trained on 10M PubMed abstracts. Some experiments in biomedical NER and NLI tasks show that BioBERT outperforms BioELMo when fine-tuned, however, when the model is used as a feature extractor, BioELMo is better than BioBERT. This phenomenon occurs because BioELMo is more effective in encoding entity types and information about biomedical relations.

We decided to fine-tune BioELMo, due to computational resources, on the MEDIQA-QA task training dataset to compute biomedical contextual embeddings for each word in the sentence. Domain-specific embeddings for the question and the answer are computed by taking the mean of the hidden states from the last LSTM layers.

Finally, the question and answer embeddings are concatenated and follow the same path, Figure 4, as the BioBERT, SciBERT, and Bio ClinicalBERT question-answer representations.

5.4 Losses

During training we make use of a combination of the Binary Cross Entropy and a Ranking Loss.

The **Binary Cross Entropy** (BCE) loss computes the error of classifying the question-answer pair as entailed or not-entailed. For a binary classification like our system, the typical used loss function is the BCE. However, since we have an imbalanced dataset ($\sim 35\%$ of positive samples and $\sim 75\%$ of negative samples) we use a weighted BCE loss.

$$loss(x, y) = -\frac{1}{N}\sum_{i=1}^{N} w_1\ y_i\ log(p(x_i)) + w_2\ (1 - y_i)\ log(1 - p(x_i)) \tag{1}$$

where y_i is the label (0 or 1), and $p(x_i)$ is the predicted probability of the answer being entailed or not, and $w_1 = \frac{75}{35} = 1.85, w_2 = 1$.

To improve the result metrics related to answer ranking (MRR, Spearman's rho) we add another loss called **Ranking loss** (RL) to force the question and answer embeddings to be more/less similar to

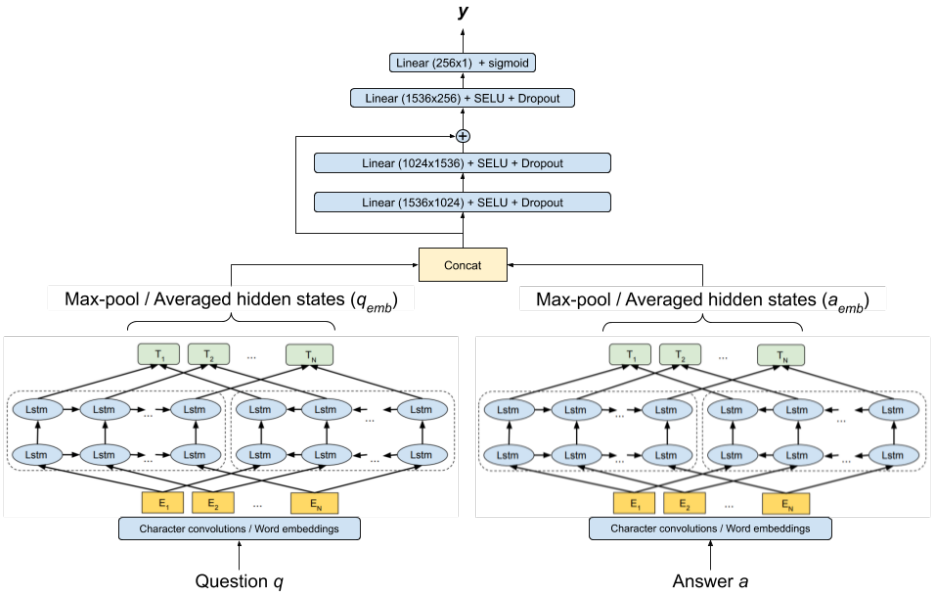


Figure 4: Model architecture using ELMo-based sentence embeddings. ELMo images retrieved from [Devlin et al., 2018].

each other depending on the ground truth ranking. For instance, given a question q_n if answer a_i is ranked before answer a_j, we are forcing that the the distance between the embeddings should be $dist(q_{n_{emb}}, a_{i_{emb}}) < dist(q_{n_{emb}}, a_{j_{emb}})$. This is achieved by setting a wide margin between positive and negative samples and narrow mini-margins between ranked positive margins, as shown in figure 5.

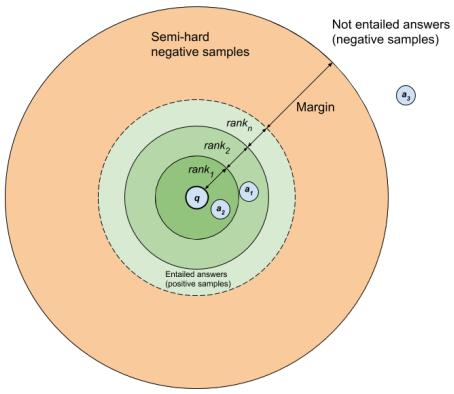


Figure 5: Representation of the margin ranking loss for positive and negative pairs of questions and answers, where a_3 is a negative sample and a_1, a_2 are positive ones.

The distance function, Equation 2, is defined by the inverse of the cosine similarity between the question embedding and the answer embedding. Equation 3 shows the RL loss.

$$dist(q_{emb}, a_{emb}) = 1 - CosineSimilarity(q_{emb}, a_{emb}) \tag{2}$$

Note the this distance metric ranges from 0 to 1, hence, the Ranking Loss function for a batch of samples is defined as in equation 3.

$$RankLoss(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y}) = \frac{1}{N}\sum_{i=1}^{N} y_i \cdot (\text{Mini\_Margin} - dist(q_{i_{emb}}, a_{i_{emb}}))^2 + (1-y) \cdot (\text{Margin} - dist(q_{i_{emb}}, a_{i_{emb}}) \tag{3}$$

where Margin = 0.75 and Mini_Margin = $0.005 \cdot (a_{rank} - 1)$, and a_{rank} is the ground truth rank of the answer. Note that the greater a_{rank}, the greater the mini-margin. Finally, the final training loss is defined in Equation 4 as a weighted sum of the *WeightedBCE* and the *RankLoss*:

$$loss(x, y) = \lambda_1 \cdot WeightedBCE(p(x), y) + \lambda_2 \cdot RankLoss(q_{emb}, a_{emb}, y) \tag{4}$$

Where λ_1 and λ_2 are weighting parameters, in our case $\lambda_1 = 0.4$, $\lambda_2 = 0.6$.

5.5 Answers classification and ranking

Given a medical question our QA system should be able to perform two different tasks:

1. Classify the answers related to the question as entailed or not, label 0 means no relation, and label 1 the opposite.
2. Give a ranking of answers to the question from more appropriate to less.

For a pair of question-answer, the first system objective is achieved by taking those output probabilities greater than a specific threshold. Usually, the value of the threshold is set to 0.5. For a specific question, answers that have probabilities bigger than 0.5 are classified as entailed (label 1), whereas those that are less than 0.5 mean that they don't have any relation with the given question (label 0).

For ranking answers from more accurate to less related to a question we can follow two approaches:

- **Ranking by y Label output**: we sort in a decreasing way the output probabilities greater than 0.5. Therefore, given a question, the answer with the biggest probability is ranked in the first position and the lowest is ranked in the last position.
- **Ranking by embedding distance**: if ranking loss has significance in the loss computation (λ_2 parameter is large), we can use the cosine distance between the question ans answer embeddings to rank the answers for a given question. Since the ranking loss forces embedding distances to be larger for less entailed answers, it makes sense to use it for predicting a better ranking.

6 Experiments and Results

In this section we compare BioBERT, Bio ClinicalBERT and BioELMo pre-treinde models and report their evaluation metric scores. Note that different architectures setups have been used during the experiments. BioBERT, SciBERT and Bio ClinicalBERT use the same structure and BioELMO uses a different one.

6.1 Data

We used the MEDQA19 Challenge datasets for these experiments which are described in section 3.2. We recall that MEDIQA19-QA dataset contained 208, 25 and 150 questions associated with a ranked

list of answers for each of the training, validation and test sets, respectively. If we reassemble these dataset to build question-answer pairs for feeding the model, we and up having datasets of 1701, 234 and 1107 samples for each the train, validation and tests sets, respectively. Also, as we mentioned before, we benefit from the MedQuAD additional dataset to by adding 71960 new data samples.

6.2 Experimental setups

We trained the BioBERT, Bio ClinicalBERT, SciBERT, Vanilla BERT, and BioELMo pre-trained models on the MEDIQA 2019 Task 3 training dataset. The experiments were run on an NVIDIA GeForce RTX 2060 GPU of 6GB VRAM, which is quite limited regarding the model and dataset size (when adding external resources). These modest computational resources limited us to train for few epochs and only fine-tune few layers of the pre-trained models.

For fine-tunning, the maximum sequence length was set to 512, a batch size of 8 was selected and a learning rate of 5e-5 was selected. We fine-tune the last 3 layers of the encoder in the case of BERT models (our GPU RAM was not able to handle more parameters). Fine-tunning the models for the QA task took up to an hour just training with small training data provided by the shared-task Challenge, and almost a whole day with the external synthetic dataset from MedQuAD.

The training procedure is to use the validation set to control model overfitting with early stopping of 5 epochs of patience over the improvement of validation loss. Also, we recover the model with better validation accuracy and perform each experiment 3 times and take the best model achieved out of them.

6.3 Experimental results

In this section, we first demonstrated potential use of the Margin Ranking Loss metric in tasks where answers should be ranked from more to less appropriate. We also include a benchmark of the biomedical-based pre-trained models with different setups and training dataset sizes.

For the first experiments related to model architecture, we only train with the training dataset provided by MEDQA19 that consists of 1701 ($\sim$ 2k) data samples, which take from 30 minutes to 1 hour of training on average before early stopping. We choose to first train with the official training dataset because of our limited resources, and also we were unsure if our extended synthetic data from MedQuAD will help the model.

6.3.1 BioBERT, BioClinicalBERT, SciBERT, Vanilla BERT and BioELMo

We compared different pre-trained encoders with the same training dataset and setup, training loss weightings to $\lambda_1 = \lambda_2 = 0.5$ and CLS token as embedding strategy.

Model	**Embedding strategy**	**Accuracy**	**Precision**	**MRR**	**Spearman's rho**
BioBERT-1.1	CLS token	0.564	0.595	0.628	0.017
BioClinicalBERT	CLS token	0.562	0.586	0.642	0.027
SciBERT-nli	CLS token	0.566	0.595	0.680	0.012
Vanilla BERT	CLS token	0.555	0.578	0.628	-0.019
BioELMo	Mean pooling	0.533	0.543	0.553	0.079
Provided Answers	-	*0.517*	*0.516*	*0.895*	*0.315*

Table 2: Model performance trying different pre-trained encoders.

Test results of the QA task are shown in Table 2. We note that there is not a significant difference between BioBERT, Bio ClinicalBERT, SciBERT, and even BERT still manages to reach comparable results, although not being trained with medical corpora. Maybe with more training data, this comparison between BERT-based models would have shown more differences.

On the other hand, the BioELMo model reached worse accuracy, and, most important, it was tremendously slow at training with our computational settings. That's why we are not considering it for our final model.

In this case, we decide to chose BioClinicalBERT since it is a BioBERT model fine-tuned on clinical notes. This means that Bio ClinicalBERT will enrich the model as it would have seen both biomedical and clinical corpora during training, therefore, the QA system, could take advantage as it deals with all types of medical questions. Hence, from now on we will consider Bio ClinicalBERT in our model architecture.

6.3.2 Margin Ranking Loss

We observe the behavior of using the ranking loss by changing the weights λ_1 and λ_2, which affect the contribution of the binary cross-entropy and the ranking loss to the total loss of the model. In Table 3, we can observe that giving significant weight to the ranking loss helps to improve ranking metrics (MRR and Spearman's rho) while losing slightly on accuracy and precision.

Model	Loss weighting	Accuracy	Precision	MRR	Spearman's rho
Bio ClinicalBERT CLS	$\lambda_1 = 1 \quad \lambda_2 = 0$	**0.556**	**0.586**	0.650	0.017
Bio ClinicalBERT CLS	$\lambda_1 = 0.2$ $\lambda_2 = 0.8$	0.552	0.566	**0.706**	**0.140**

Table 3: Results using Margin Ranking Loss. When ranking loss has significant weight (λ_2 set to 0.8), we get significant improvement on ranking metrics like MRR and Spearman's rho by comparing distance of the embeddings.

From now on, we found it beneficial to set $\lambda_1 = \lambda_2 = 0.5$ since the important metric is the accuracy, and giving too much importance to ranking loss rather than to the binary cross-entropy can sometimes lead to poor model performance.

6.3.3 Embedding strategy

In this section we evaluate which embedding strategies are the most suitable for our problem: using the CLS token, averaging the hidden token states (mean pooling), or performing a max pooling on the hidden token states. Table 4 shows that all strategies perform well but max-pooling seems to succeed at most of the metrics. Moreover, max pooling is commonly used in many other works related to sentence embedding [Zhou et al., 2019, Artetxe and Schwenk, 2019] and appears to be a safe choice [Chen et al., 2018]. For all these reasons, we are going to opt for the max-pooling strategy for our embeddings model.

Model	Embedding strategy	Accuracy	Precision	MRR	Spearman's rho
Bio ClinicalBERT	CLS token	0.562	0.569	**0.698**	0.039
Bio ClinicalBERT	Mean pooling	**0.571**	0.581	0.519	-0.185
Bio ClinicalBERT	Max pooling	**0.571**	**0.607**	0.342	**0.205**
Provided Answers	-	*0.517*	*0.516*	*0.895*	*0.315*

Table 4: Results obtained trying different sentence embedding strategies.

6.3.4 Training with external data

As we explained before, we enrich the training dataset by building question-answer samples from the MedQuAD corpus. We only train the BioClinicalBERT model with this amount of data since it becomes is computationally hard to fine-tune the model.

Each epoch took around 4 hours to train with the whole extra dataset, thus, the whole model took almost 1 day to train. In table 5, we can see the improvement of training with many more training samples and that the model can capture more language understanding.

Model	**Number of training samples**	**Accuracy**	**Precision**	**MRR**	**Spearman's rho**
Bio ClinicalBERT	∼2k	0.571	0.607	0.342	0.205
Bio ClinicalBERT	∼70k	**0.634**	**0.668**	**0.768**	**0.066**
Provided Answers	-	*0.517*	*0.516*	*0.895*	*0.315*

Table 5: Improvement achieved when using external data.

6.4 Our best model vs top-10 challenge participants

The Bio ClinicalBERT pre-trained model that obtained the best results is the one with loss weighting of 0.5, max pooling as embedding strategy and trained on ∼70k samples. We managed to implement a model that beats the provided baseline (ChiQA), and achieves the 8th position among the top 10 participants of the MEDIQA 2019 QA challenge as seen in Table 6. We can observe that Spearman's rho is the most unstable metric, and we are unable to get a good score.

Rank	**Team**	**Accuracy**	**Precision**	**MRR**	**Spearman's rho**
1	DoubleTransfer	0.780	0.8191	0.9367	0.238
2	PANLP	0.777	0.7806	0.9378	0.180
3	Pentagon	0.765	0.7766	0.9622	0.338
4	DUT-BIM	0.745	0.7466	0.9061	0.106
4	DUT-NLP	0.745	0.7466	0.9061	0.106
6	IITP	0.717	0.7936	0.8611	0.024
7	lasigeBioTM	0.637	0.5975	0.91	0.211
-	**Ours**	**0.634**	**0.668**	**0.768**	**0.066**
8	ANU-CSIRO	0.584	0.5568	0.7843	0.122
9	Dr.Quad	0.565	0.6679	0.6069	0.009
10	ARS NITK	0.536	0.5596	0.6293	0.196
-	*Provided Answers*	*0.517*	*0.5167*	*0.895*	*0.315*

Table 6: Official results of the top-10 participants of the MEDIQA19 QA Task with our proposed method ranked too.

7 Discussion

In this project, we compared different variations of BERT and ELMo encoders to solve the MEDiQA 2019 task 3. BioBERT, SciBERT, Bio ClinicalBERT, and BioELMo are pre-trained language representation models for biomedical, scientific, or/and clinical text mining. We have seen that using pre-trained BERT models in the biomedical domain gives slightly better metric results than BERT,

however, this might be because we used a small dataset during training. Training with larger models would probably have improved the results significantly. Besides, it would have strengthened the difference between the BERT base and other biomedical variations. We discarded BioELMo as it obtained the worst results probably for our limited resources. We decided to use Bio ClinicalBERT as gives good results, and as it was pre-trained above BioBERT it would have seen more task-related context information.

Unlike other loss functions, such as Cross-Entropy, whose objective is to learn to predict directly a label, the objective of ranking losses is to predict relative distances between inputs. Here we had to use a combination of a weighted Binary Cross-Entropy, for answer classification, and Margin Ranking Loss, for answer ranking. As shown in Table 3, we analyzed the contribution of the weighted BCE loss and Margin Ranking Loss (MRL) metrics. We have found empirical evidence that our custom ranking loss (MRL) increases ranking evaluation scores (MRR and Spearman's rho), meaning that it can be useful for retrieving better-ranked answers.

Sentence embedding is an important research topic in NLP since it is used to transfer knowledge to downstream tasks such as QA. It is an open problem to generate high-quality sentence representations from BERT (or its variations). Previous studies have shown that different layers of BERT capture different linguistic properties. For this reason, we studied three embedding strategies for question-and-answer sentences. Our results suggested that the best strategy is to use a max-pooling of hidden layers of all token embedding produced by the model. However, for future research could be interesting to explore other strategies like SBERT-WK [Wang and Kuo, 2020], this sentence embedding approach uses geometric analysis of the space learned by deep contextualized models. A method like SBERT-WK may leverage more diverse information learned in different layers to produce more effective QA representations.

We used corpora of different sizes for training Bio ClinicalBERT and investigate their effect on performance. Table 5 shows the results against the number of training samples. The performance of Bio ClinicalBERT with $\sim$ 70k samples outperforms the model trained with $\sim 2k$. These results indicate that training with larger models improves the results, and, more interestingly, that our synthetic dataset helps to obtain better performance.

8 Conclusion

In conclusion, despite limitations on computational resources our best model obtained the 8th position among top-10 participants in the MEDIQA 2019 QA Task. We remark that this challenge is quite difficult since best reported accuracy is 0.78. However, these absolute results demonstrate that our strategy is not that far from the top promising results. Top participants used multi-task language models and ensable methods taking advantage of the information extracted from Natural Language Inference and Contextual Entailment tasks.

References

[Andrenucci, 2008] Andrenucci, A. (2008). Automated question-answering techniques and the medical domain. In *HEALTHINF (2)*, pages 207–212.

[Artetxe and Schwenk, 2019] Artetxe, M. and Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond.

[Beltagy et al., 2019] Beltagy, I., Lo, K., and Cohan, A. (2019). Scibert: A pretrained language model for scientific text.

[Ben Abacha and Demner-Fushman, 2019] Ben Abacha, A. and Demner-Fushman, D. (2019). A question-entailment approach to question answering. *arXiv e-prints*.

[Ben Abacha et al., 2019] Ben Abacha, A., Shivade, C., and Demner-Fushman, D. (2019). Overview of the MEDIQA 2019 shared task on textual inference, question entailment and question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 370–379, Florence, Italy. Association for Computational Linguistics.

[Chen et al., 2018] Chen, Q., Ling, Z.-H., and Zhu, X. (2018). Enhancing sentence embedding with generalized pooling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1815–1826, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

[Demner-Fushman et al., 2019] Demner-Fushman, D., Mrabet, Y., and Ben Abacha, A. (2019). Consumer health information and question answering: helping consumers find answers to their health-related information needs. *Journal of the American Medical Informatics Association : JAMIA*, 27.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Green Jr et al., 1961] Green Jr, B. F., Wolf, A. K., Chomsky, C., and Laughery, K. (1961). Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224.

[Huang et al., 2020] Huang, K., Altosaar, J., and Ranganath, R. (2020). Clinicalbert: Modeling clinical notes and predicting hospital readmission.

[Jin et al., 2019] Jin, Q., Dhingra, B., Cohen, W. W., and Lu, X. (2019). Probing biomedical embeddings from language models. *arXiv preprint arXiv:1904.02181*.

[Kupiec, 1993] Kupiec, J. (1993). Murax: A robust linguistic approach for question answering using an on-line encyclopedia. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 181–190.

[Lee et al., 2019] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2019). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*.

[Liu et al., 2019] Liu, X., He, P., Chen, W., and Gao, J. (2019). Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

[Simmons et al., 1964] Simmons, R. F., Klein, S., and McConlogue, K. (1964). Indexing and dependency logic for answering english questions. *American Documentation*, 15(3):196–204.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., and Polosukhin, I. (2017). Attention is all you need. *In Advances in neural information processing systems*, pages 5998–6008.

[Wang and Kuo, 2020] Wang, B. and Kuo, C.-C. J. (2020). Sbert-wk: A sentence embedding method by dissecting bert-based word models. *arXiv preprint arXiv:2002.06652*.

[Zhou et al., 2019] Zhou, H., Li, X., Yao, W., Lang, C., and Ning, S. (2019). DUT-NLP at MEDIQA 2019: An adversarial multi-task network to jointly model recognizing question entailment and question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 437–445, Florence, Italy. Association for Computational Linguistics.