

ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ



ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

School of Information Sciences and Technology
Department of Informatics
Athens, Greece

Master Thesis
in
Computer Science

Biomedical Question Answering

Petros Stavropoulos

Supervisors: Ion Androutsopoulos

Dimitris Pappas

November 2020

Abstract

Question Answering and Machine Reading Comprehension (MRC) are crucial and complex tasks in the field of Natural Language Processing (NLP). In this thesis, we first introduce BioMRC, a novel biomedical dataset for cloze-type Question Answering, based on previous work of the BioRead dataset, implementing the same baselines and models for comparison. We then develop two new models based on the SciBert model from AllenAI for solving the task of BioMRC. We use these pre-trained models as a transfer learning approach for the BioASQ Task 8B Phase B, in a modified architecture, to investigate whether our dataset can be used for improving exact answer Question Answering tasks. In addition, we experiment with other BERT-based models for solving the BioASQ task, which use the SpanBert and BioBert models, as well as the Text-to-Text Transfer Transformer (T5) model, a generative Transformer-based model, which achieved the best results for the task. Moreover, we create a cloze-type version of the BioASQ Task 8B Phase B factoid instances subset, which is used to boost the T5's results when pre-trained on the BioMRC dataset, but can also be used in future work for automatic transformation of question-answer instances to cloze-type question instances. Lastly, we perform error analysis of our best model for the BioASQ task for exact answers, where we point out the shortcomings of the task evaluation measures and some mistakes, that could be fixed by the BioASQ organizers, as an improvement of the task.

Περίληψη

(Machine Reading Comprehension) (Question Answering) (Natural Language Processing),
BioMRC, BioRead. BioRead SciBert
AllenAI task BioMRC. BioMRC, BioASQ Task 8B Phase B,
(transfer learning) (exact answer). SpanBert BioBert
BioASQ Text-to-Text Transfer Trans-
former (T5), (spans)
BioMRC, (factoid questions)
(error analysis) BioASQ
BioASQ,

Acknowledgements

I would like to express my sincere gratitude and thanks to my supervisor Prof. Ion Androutsopoulos for his constant support, encouragement and guidance for the completion of this thesis. I would also like to thank Dimitris Pappas for his invaluable advice, cooperation and share of knowledge, which made the process of researching in this field a truly enjoyable task.

In addition, I would like to thank all members of the AUEB NLP Group, who not only provided me with fruitful ideas for the field but also with the skills needed to review and question the latest research work done in the field, for the interest of improving it. Lastly, I would like to thank my family and all my friends, who supported and encouraged me, through all this time.

Contents

Abstract	iii
Acknowledgements	iv
1 Introduction	1
1.1 Overview	2
1.2 Importance	3
1.3 Accomplishments	4
2 The BioMRC dataset and previous methods	6
2.1 BioRead Problems	6
2.2 Pubtator web tool	7
2.3 Construction of the dataset	7
2.4 Implementing previous MRC models	8
2.5 BioMRC examples	11
3 SciBERT, new methods and evaluation	13
3.1 BERT-SciBERT overview	13
3.2 BERT problems with max sequence size	14
3.3 SciBERT as Embedding for ASReader - AOARReader	15
3.4 SciBERT models	15
3.4.1 Multiplication, BiLinear multiplication with Logistic Regression	16
3.4.2 SciBERT and Multi Layer Perceptron (MLP)	16
3.5 Results of SciBERT-Reader models	18
3.6 Human evaluation and MRC models comparison	19
4 Transfer learning to BioASQ	22
4.1 Evaluation Measures of BioASQ Task 8B Phase B	22
4.2 Zero-shot using SciBERT-Reader models	23
4.2.1 Pre-trained SciBERT-Reader model approach	23
4.3 SpanBERT, BioBERT approaches	25
4.4 Text-to-Text Transfer Transformer (T5) approach	27
4.5 BioASQ in BioMRC format	29
4.6 Results and problems of BioASQ	29

5 Conclusions and future work	34
5.1 Overview	34
5.2 Accomplishments	35
5.3 Extra contributions	36
5.3.1 Covid-19 Search Engine	37
5.3.2 BioASQ 8 Challenge	38
5.4 Future work	38
Bibliography	40
List of Figures	44
List of Tables	46

Introduction

In this MSc thesis we will present the work and all the experiments that have been conducted during the work of the thesis in the domain of the Biomedical Machine Reading Comprehension (MRC) (Hermann et al., 2015). MRC is the field of Natural Language Processing (NLP) and Machine Learning where a machine learns to read and comprehend unstructured text and consequently answers questions related to the content of the text.

Many datasets are available for MRC across many domains. Some have been created using books or news articles, like the CBTest (Hill et al., 2016) and CNN Daily Mail (Hermann et al., 2015) datasets respectively. These datasets introduced a Question Answering task, where instead of answering a question directly, a word or entity from a sentence is obscured and then a model attempts to predict it. In order to do that, some sentences are given as context for the model, along with a set of candidate answers, out of which the model tries to choose the correct word or entity. This task is also referred to as a cloze-type Question Answering task (Taylor, 1953). Using this cloze-type format, we can get automatically generated Question Answering instances from a corpus. This solves a big issue, as datasets like SQUAD (Rajpurkar et al., 2016), which use human annotators for Question Answering are very costly.

This format was also adopted by many research teams in order to create cloze-type Question Answering datasets in the biomedical domain. An example of this is CLICR (Šuster and Daelemans, 2018), which was created using full-text articles from BMJ case reports¹. The dataset contained 100k instances of passages and questions and a tool named CLAMP (Soysal et al., 2017) was applied in order to detect biomedical entities in the text and subsequently link them to concepts of the Unified Medical Language System (UMLS) (Lindberg et al., 1993) metathesaurus. The cloze-type questions were created by replacing biomedical entities that were considered as 'learning points' (summarized important information of a given article) with placeholders.

Another cloze-type Question Answering dataset in the biomedical domain, which was crucial for our work, is the BioREAD dataset (Pappas et al., 2018). The dataset was created using 90.6k full-text articles from PubMed Central², producing approximately 16.4 million question instances. In order to create the instances, the text is extracted from the HTML PubMed Central articles and the MetaMap (Aronson and Lang, 2010) annotation

¹<https://casereports.bmj.com/>

²<https://www.ncbi.nlm.nih.gov/pmc/>

tool is executed to get all words and phrases (biomedical entities) that link to the UMLS metathesaurus. The text is extracted into sequences of 21 sentences. The first 20 sentences are considered the context of the instance, which the model reads and therefore learns the contextual knowledge that it needs to answer the instance question. The last sentence is considered the question of the instance, from which a biomedical entity (which was annotated by MetaMap) is replaced by a placeholder. The models then attempt to find which was the original entity in the question sentence. The candidate entities are all the unique biomedical entities present in the 20 context sentences. This dataset is the main inspiration of our work, since our original goal was to improve it and dispose of its problems, which we will analyze in the next chapter.

Besides the cloze-type Question Answering datasets, many more datasets with questions which are answered by spans in the text exist and are very important for Question Answering and MRC. SQUAD (Rajpurkar et al., 2016) is a Question Answering dataset, which was created using a set of Wikipedia articles. From these articles, segments of text have been extracted and a question was created for each segment, which is then addressed by crowd-workers. The crowd-workers either answered by choosing the span of the text that contained the answer, or labeled the question as unanswerable. In the biomedical domain and crucial for our work in this thesis, is the dataset of the BioASQ Task 8B Phase B (Tsatsaronis et al., 2015) challenge. For the task, the participants are given instances that contain a question and snippets that are relevant to that question. Then a model must generate an answer, which answers the question using these snippets. The answers are meant to be biomedical entities and short in length, but as we will discuss later in this thesis, this is not always the case. Furthermore, the questions are distinguished between 'factoid', 'list', 'yes/no' and 'summary'. For the 'factoid' questions there is a single answer, whereas in the case of 'list' questions there are multiple answers. In the case of 'yes/no' instances, an affirmative or negative answer must be provided and for the 'summary' instances, a longer answer is required for a given question. A longer answer is also required for the other question types besides 'summary' questions, as an 'ideal answer' for the questions. During the work of this thesis, we participated in the BioASQ Task 8B Phase B challenge, but as we will mention later in the thesis, the results are mixed and further inspection and experimentation is needed in order for the task to be solved.

1.1 Overview

In this thesis we have created a new dataset in the biomedical domain for MRC, inspired from the BioRead dataset, which we call BioMRC (Pappas et al., 2020b). BioMRC was created to overcome the shortcomings of the BioRead dataset and improve its clarity and correctness. In terms of the models we developed in order to solve the task, we

implemented the baselines and neural models that were used in the BioRead dataset and introduced two new BERT-based models that we will analyze in the following chapters.

Furthermore, we focused on solving the BioASQ Task 8B Phase B task using transfer-learning from other Question Answering tasks including BioMRC. We experimented with a zero-shot learning technique, by using the knowledge that the models learned from the BioMRC task, as well as by fine-tuning the models directly on the BioASQ dataset. For the fine-tuning we experimented with many different BERT-based (Devlin et al., 2019) models like SciBERT (Beltagy et al., 2019), SpanBERT (Joshi et al., 2020) and BioBERT (Lee et al., 2019), as each one has different qualities that could benefit the models, when solving the task.

Moreover, we used the Text-to-Text Transfer Transformer (T5) model (Raffel et al., 2020), experimenting by pre-training it on various datasets and fine-tuning it on the BioASQ dataset. This model was our best approach for the BioASQ Task 8B Phase B task. We also created a modified version of the BioASQ dataset, to investigate whether changing the format of the question could be beneficial in the case that we use the BioMRC task as a pre-training dataset.

Lastly, we performed error analysis on the BioASQ dataset and inspected the evaluation measures that it used. This was done, as we kept getting bad results on the BioASQ task, regardless of how good our models seemed to perform. This raised questions on the validity of some of the dataset's gold answers, as well as the evaluation measures which were used to evaluate the predictions of the models.

1.2 Importance

The work presented in this thesis is important for the MRC field, as limited work has been done on biomedical question answering. The new dataset could be used by many research teams, as a pre-training step to improve on down-stream tasks in the Question Answering biomedical domain. Additionally, it is a very good benchmark task for the evaluation of MRC models, regarding the cloze-type questions.

Another important issue, that we are going to discuss, is whether we can use transfer learning to benefit from a cloze-type questions task like BioMRC, for solving a task with open questions like the BioASQ exact answers task. This would be crucial for the MRC field, as creating an annotated dataset for open questions is cumbersome and expensive, whereas a cloze-type Question Answering dataset can be created automatically with little to no cost.

Lastly, trying to solve the BioASQ exact answer task, which is answering open questions created by humans, by using retrieved snippets as context, is of vital importance. This is true, as having a model that can answer biomedical-related questions accurately and consistently can be beneficial both for people in need of info about a medical issue and also for medical professionals and researchers, who need answers for potential medical situations with insight from all medical records and peer-reviewed articles, without the hustle of going through all of them by hand.

1.3 Accomplishments

In this thesis, the contributions to the MRC field in the biomedical domain that were accomplished are the following. Firstly, a new large biomedical dataset for cloze-type Question Answering was created using an automated procedure. Along with the dataset, a human evaluation of a small part of the dataset was provided, so that models can be trained and compared to human performance in Question Answering.

Moreover, in order to solve the BioASQ Task B Phase B, a zero-shot learning model was introduced, which used the pre-trained model, on which we run our experiments for the BioMRC task. This was used, to investigate whether the two tasks were close enough contextually, so that the knowledge from the BioMRC task could be used directly in order to solve the BioASQ task. Furthermore, more BERT-based models were used in the same architecture, which were, optionally, pre-trained on BioMRC and then trained on the train set of BioASQ Task B Phase B, to test if they could solve the task better. Although a big boost in performance was not achieved, the above transfer learning method had an impact on the results, suggesting that if the BioASQ task was more contextually close to the BioMRC task, a bigger gain could be obtained.

In addition to the above approaches, the T5 model model was used in one of our models to solve the BioASQ task, using sequence generation. This approach achieved the best results amongst all experiments, which is noteworthy, considering the fact that the exact answers were generated and were not selected as a span of the snippets.

Another contribution of this thesis is the creation of a cloze-type version of the BioASQ factoid exact answer instances, in the format of the BioMRC dataset. This could help the models to make better use of the knowledge from a cloze-type Question Answering task to answer the BioASQ questions. This dataset could also be used along the original BioASQ as a dataset for the automatic reformation of question-answer tuples to a corresponding cloze-type question instance.

Lastly, the validity of the BioASQ task evaluation measures was investigated. In this process, inconsistencies and errors were found in the evaluation measures, as well as the gold answers of the dataset, which ignored the correct predictions made by the models, finding them mistakenly as false positives. To combat this issue, custom evaluation measures were tested using the fuzzywuzzy python module³, which did not force the models to answer exactly the gold answer that was given by the humans answering the question. Instead, any answer that was close enough to the gold answer was considered as a true positive. The experimental results using this evaluation measure, showed that the answers that were now predicted as true positives, were indeed correct.

³<https://github.com/seatgeek/fuzzywuzzy>

The BioMRC dataset and previous methods

In this chapter we are going to discuss the construction of the new version of the BioRead dataset (Lee et al., 2019), which we called BioMRC (Pappas et al., 2020b), as well as the implementation and results of the same neural models that were present also in the work of BioRead. The new dataset overcomes the problems that we encountered in BioRead, producing a "cleaner" version of it. The implementation of the same models as in BioRead can provide us with a comparative analysis of the results, which can indicate the "cleanliness" and the difference of the difficulty, between the two datasets.

2.1 BioRead Problems

The BioRead (Pappas et al., 2018) dataset had some problems with noise, due to the process of extracting the text from the HTMLs of PubMed¹ articles, as well as from the annotation tool MetaMap. The latter created noise and confusion to both human evaluators and the models that tried to solve the Question Answering task.

The problems pertained mainly to references, captions of figures or footnotes of the article that were inside questions or passages. These introduced noise, usually by creating a breach in the continuity of the sentence's context and thus, the information that it was pointing to was not accessible to the model.

Another source of noise was coming from the annotation tool MetaMap (Leaman et al., 2013), which was trained to annotate the biomedical entities in the articles. Even though, this tool was considered the state-of-the-art at that time, its results were not that great, as it often missed or erroneously identified biomedical entities.

Some effort was given in order to try to minimize the noise from the above two factors, using regular expressions, but it was time consuming and we did not have any way to evaluate whether the problem has been solved. In addition, MetaMap is an aging tool that needs many computational resources to run, making the above process even more difficult.

¹<https://pubmed.ncbi.nlm.nih.gov/>

2.2 Pubtator web tool

For the reasons stated in the previous section, we decided that it would be better to create a new dataset from the same domain, but at the same time eliminate the aforementioned problems from the beginning. The main idea was to use the PubMed abstracts instead of the full text, as they do not have any captions, references, footnotes or other tags that produce noise and use the titles of the articles as the cloze-type question. Instead of using MetaMap, we used a repository called PubTator (Wei et al., 2012a) which has annotated many of the PubMed abstracts using the state-of-the-art tools GeneTUKit (Huang et al., 2011), GeNorm (Vandesompele et al., 2001), DNORM (Leaman et al., 2013), SR4GN (Wei et al., 2012b), tmVar (Wei et al., 2013) and a dictionary for MESH terms (Lipscomb, 2000).

These abstracts and titles were then processed in order to create cloze-type passage-question instances, like in BioRead. In these instances, the biomedical entities from PubTator are replaced with special entity tokens with local-scope (document-specific) or global-scope (dataset-wide) identifiers. In the question, which in our case is the title, an entity token is obscured and replaced with a 'XXXX' token. The task of the dataset is to predict which of the entity tokens in the passage matches the 'XXXX' token.

2.3 Construction of the dataset

In order for the dataset to be valid and for its task not to be trivial to solve, specific rules must be followed when constructing the dataset instances. All the instances that do not follow these rules must be discarded, so that the dataset is clean and the task does not have any invalid instances, which can confuse the model that tries to solve it.

From PubTator we gathered approximately 25 million abstracts, with their corresponding titles and annotations. We discarded articles with titles shorter than 15 characters or longer than 60 tokens, articles without abstracts, or with abstracts shorter than 100 characters, or fewer than 10 sentences. We also removed articles with abstracts containing fewer than 5 entity annotations, or fewer than 2 or more than 20 distinct biomedical entity identifiers. We also discarded articles containing entities not linked to any of the ontologies used by PubTator, or entities linked to multiple ontologies (entities with multiple ids), or entities whose spans overlapped with those of other entities. We also removed articles with no entities in their titles, and articles with no entities shared by the title and abstract. Finally, we removed all instances in which the answer was the most frequent entity in the passage, in order to make the dataset harder. If multiple entities had the same top frequency in the passage, the instance was not discarded though, as a model like the above would have to choose one of the most frequent entities randomly.

After applying all the rules, we ended up with approximately 812k passage-question instances, composing the BioMRC Large dataset. Because there are many research teams with limited computational resources, we created a subset of the large dataset, with only 100k instances, called the BioMRC Lite dataset.

As already stated, the biomedical entities of the passage and the title, which were annotated by PubTator, were replaced by special entity tokens in the format '@entityN' where 'N' is a number indicating the id of the entity. The dataset was constructed using the same two settings as in the BioRead dataset. Setting A uses a global scope, meaning that each entity has a unique id throughout the dataset, which corresponds to a specific biomedical entity. This setting allows a model to learn information about each entity from all the mentions of the entity in the entire dataset. Setting B on the other hand, uses a local scope, meaning that each entity has a unique id locally in each passage. This setting forces a model to comprehend the text and answer the question using only the local context of the entities, instead of learning what each entity token means from the entire dataset, like in setting A.

Each version of the dataset and each setting is evaluated separately and the results of the models are provided in the next sections of the thesis. The dataset was first split into train-dev-test sets and the statistics regarding each set are reported in the following tables.

	BioMRC Large			
	Training	Development	Test	Total
Instances	700,000	50,000	62,707	812,707
Avg candidates	6.73	6.68	6.68	6.72
Max candidates	20	20	20	20
Min candidates	2	2	2	2
Avg abstract len.	253.79	257.41	253.70	254.01
Max abstract len.	543	516	511	543
Min abstract len.	57	89	77	57
Avg title len.	13.93	14.28	13.99	13.96
Max title len.	51	46	43	51
Min title len.	3	3	3	3

Tab. 2.1: Statistics of BioMRC Large. All lengths are measured in tokens using a whitespace tokenizer.

2.4 Implementing previous MRC models

In the work for the BioRead dataset, the neural models Attention-Sum Reader (ASReader) (Kadlec et al., 2016) and Attention-Over-Attention Reader (AOARReader) (Cui et al., 2017)

	BioMRC Lite			
	Training	Development	Test	Total
Instances	87,500	6,250	6,250	100,000
Avg candidates	6.72	6.68	6.65	6.71
Max candidates	20	20	20	20
Min candidates	2	2	2	2
Avg abstract len.	253.78	257.32	255.56	254.11
Max abstract len.	519	500	510	519
Min abstract len.	60	109	103	60
Avg title len.	13.89	14.22	14.09	13.92
Max title len.	49	40	42	49
Min title len.	3	3	3	3

Tab. 2.2: Statistics of BioMRC Lite. All lengths are measured in tokens using a whitespace tokenizer.

	BioMRC Tiny		
	Setting A	Setting B	Total
Instances	30	30	60
Avg candidates	6.60	6.57	6.58
Max candidates	13	11	13
Min candidates	2	3	2
Avg abstract len.	248.13	264.37	256.25
Max abstract len.	371	386	386
Min abstract len.	147	154	147
Avg title len.	14.17	14.70	14.43
Max title len.	21	35	35
Min title len.	6	4	4

Tab. 2.3: Statistics of BioMRC Tiny. The questions were answered by humans. All lengths are measured in tokens using a whitespace tokenizer.

were implemented in order to solve the task of MRC (Hermann et al., 2015), answering the cloze-type questions (Taylor, 1953). These models were at the time the state-of-the-art approaches to the task and used Gated Recurrent Units (GRUs) (Cho et al., 2014) and an attention mechanism (Bahdanau et al., 2015) to achieve their goal. The models differed mainly in the attention mechanism, as AOARReader had a more profound and complicated mechanism from ASReader. For our work in BioMRC we implemented the same models, so that we could run them on the new dataset and compare their results to those of the BioRead dataset.

The ASReader model takes an input the abstract and the title of an instance and passes them through an Embedding Layer with learnable parameters in order to get the word representations for each word in the embedded space. The word representations are then

given to the GRUs to get the contextual representations for the abstract and the title respectively. Then the dot product of the last state of the title GRU and each state of the abstract GRU is calculated, getting a score for each word in the abstract. A softmax layer is then applied to the output in order to get a probability distribution of each word in the abstract. The process described is the attention mechanism of ASReader and the probability distribution scores are the attention scores for each word in the abstract. For the final output, ASReader sums all the attention scores for each candidate entity to get their final attention scores. The answer for the instance is therefore the candidate entity with the highest final attention score.

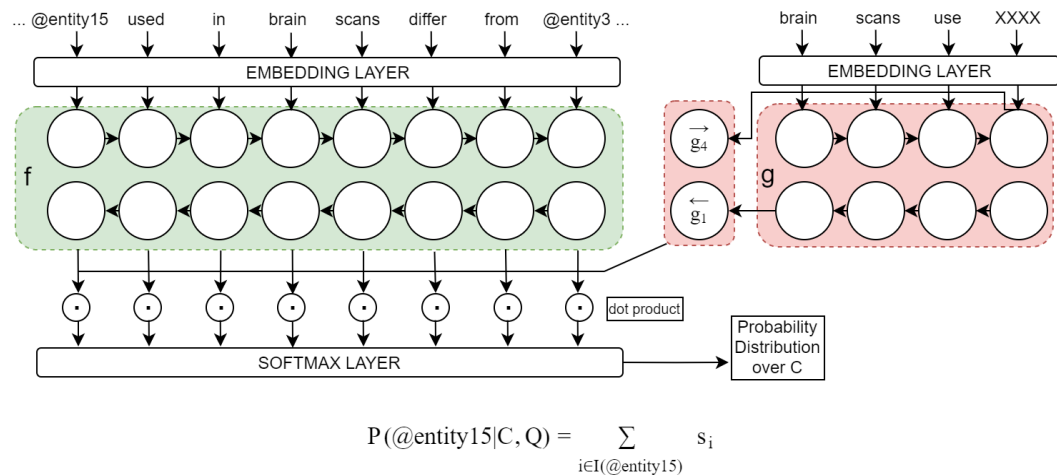


Fig. 2.1: Overview of the ASReader model.

The AOReader model is very similar to the ASReader model, changing only the attention mechanism, which is modified as following. After getting the word representation from the GRUs, the dot product of them is calculated, resulting in a pairwise scores matrix with dimensions equal to the number of words in the abstract by the number of words in the title. A column-wise and row-wise softmax is then applied separately to the above matrix, which corresponds to a query-to-document attention and document-to-query attention respectively. Then a column-wise average is applied to the matrix of the document-to-query attention to get the averaged query-level attention vector. The dot product of the query-to-document attention matrix with the averaged query-level attention vector gives the probability distribution, or attention scores, of every word in the abstract. As in ASReader, for the final attention scores, AOReader sums all the attention scores for each candidate entity to get their final attention scores and the one with the largest final attention is returned as the answer.

For the training of the models, we used the training set and fine-tuned the models' hyperparameters in the development set, whereas the test set was used for the evaluation of the final results. We trained the above models both in the Large and Lite versions of BioMRC, but because of our limited computational resources, the fine-tuning process could only be performed in the Lite version of the dataset. The hyperparameters achieving the best

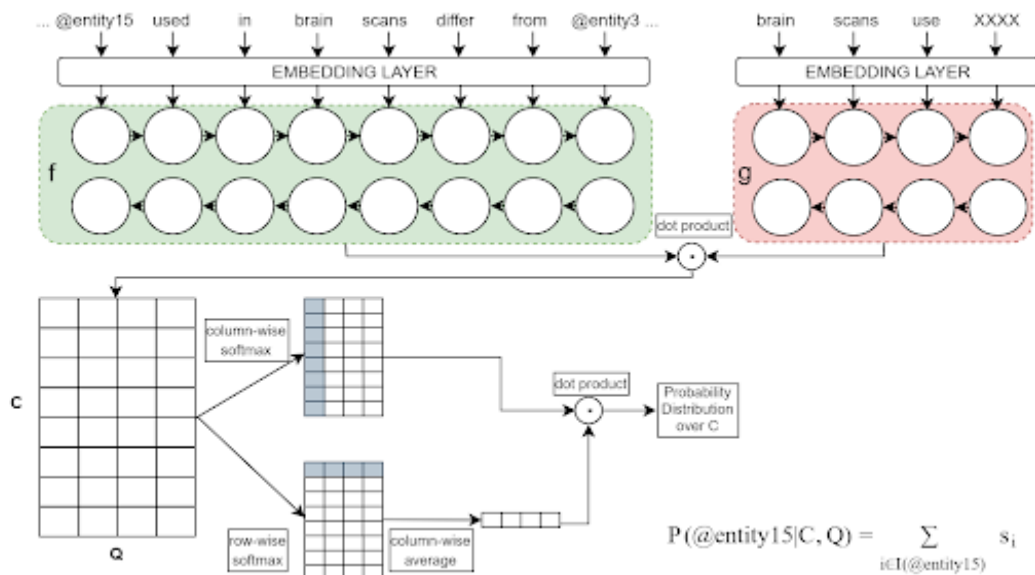


Fig. 2.2: Overview of the AOARReader model.

results in the Lite version of the dataset were then also used for the Large dataset. This means that the results on the Large dataset could be improved further, by fine-tuning on the development part of the Large dataset. For all our results we performed single-tailed significance tests ($p < 0.02$) using the method of Approximate Randomization (Dror et al., 2018). This was done in order to verify that the yielded results were not random.

2.5 BioMRC examples

In the figures below, there are two instance examples from the BioMRC dataset. The first is from the Lite version of the dataset, whereas the second is from the Tiny version of the dataset, which were also answered by human evaluators.

Passage	BACKGROUND: Most brain metastases arise from @entity0 . Few studies compare the brain regions they involve, their numbers and intrinsic attributes. METHODS: Records of all @entity1 referred to Radiation Oncology for treatment of symptomatic brain metastases were obtained. Computed tomography (n = 56) or magnetic resonance imaging (n = 72) brain scans were reviewed. RESULTS: Data from 68 breast and 62 @entity2 @entity1 were compared. Brain metastases presented earlier in the course of the lung than of the @entity0 @entity1 (p = 0.001). There were more metastases in the cerebral hemispheres of the breast than of the @entity2 @entity1 (p = 0.014). More @entity0 @entity1 had cerebellar metastases (p = 0.001). The number of cerebral hemisphere metastases and presence of cerebellar metastases were positively correlated (p = 0.001). The prevalence of at least one @entity3 surrounded with > 2 cm of @entity4 was greater for the lung than for the breast @entity1 (p = 0.019). The @entity5 type, rather than the scanning method, correlated with differences between these variables. CONCLUSIONS: Brain metastases from lung occur earlier, are more @entity4 , but fewer in number than those from @entity0 . Cerebellar brain metastases are more frequent in @entity0 .
Candidates	@entity0 : ['breast and lung cancer']; @entity1 : ['patients']; @entity2 : ['lung cancer']; @entity3 : ['metastasis']; @entity4 : ['edematous', 'edema']; @entity5 : ['primary tumor']
Question	Attributes of brain metastases from XXXX .
Answer	@entity0 : ['breast and lung cancer']

Fig. 2.3: Example passage-question instance of BioMRC. The passage is the abstract of an article, with biomedical entities replaced by @entityN pseudo-identifiers. The original entity names are shown in square brackets. Both 'edematous' and 'edema' are replaced by '@entity4', because Pubtator considers them synonyms. The question is the title of the article, with a biomedical entity replaced by XXXX. @entity0 is the correct answer.

Passage	The study enrolled 53 @entity1 (29 males, 24 females) with @entity1576 aged 15-88 years. Most of them were 59 years of age and younger. In 1/3 of the @entity1 the diseases started with symptoms of @entity1729, in 2/3 of them—with pulmonary a ection. @entity55 was diagnosed in 50 @entity1 (94.3%), acute @entity3617—in 3 @entity1. ECG changes were registered in about half of the examinees who had no cardiac complaints. 25 of them had alterations in the end part of the ventricular ECG complex; rhythm and conduction disturbances occurred rarely. Mycoplasmosis @entity1 suffering from @entity741 (@entity741) had stable ECG changes while in those free of @entity741 the changes were short. @entity296 foci were absent. @entity299 comparison in @entity1 with @entity1576 and in other @entity1729 has found that cardiovascular system sufferers less in acute mycoplasmosis. These data are useful in differential diagnosis of @entity296.
Candidates	@entity1 : ['patients']; @entity1576 : ['respiratory mycoplasmosis']; @entity1729 : ['acute respiratory infections', 'acute respiratory viral infection']; @entity55 : ['Pneumonia']; @entity3617 : ['bronchitis']; @entity741 : ['IHD', 'ischemic heart disease']; @entity296 : ['myocardial infections', 'Myocardial necrosis']; @entity299 : ['Cardiac damage'].
Question	Cardio-vascular system condition in XXXX.
Expert Human Answers	annotator1: @entity1576; annotator2: @entity1576.
Non-expert Human Answers	annotator1: @entity296; annotator2: @entity296; annotator3: @entity1576.
Systems' Answers	: @entity1729; : @entity296; : @entity1576.

Fig. 2.4: Example from BioMRC Tiny. In Setting A, humans see both the pseudo-identifiers (@entityN) and the original names of the biomedical entities (shown in square brackets). Models see only the pseudo-identifiers, but the pseudo-identifiers have global scope over all instances, which allows the models, at least in principle, to learn entity properties from the entire training set. In Setting B, humans no longer see the original names of the entities, and models see only the pseudo-identifiers with local scope (numbering reset per passage-question instance).

SciBERT, new methods and evaluation

In this chapter we are going to investigate additional methods to solve the task for the dataset which we presented in the previous chapter. All the methods are based on the BERT model (Devlin et al., 2019), as it was proved to be a very useful contribution in the NLP field, capable of solving a variety of tasks, providing the state-of-the-art results and can be transferred and fine-tuned to many other NLP tasks with little or no training at all. This is particularly important to us, as we can implement and train a model, in spite of our limited computational resources.

Here we will provide an overview of the BERT model, a fine-tuned version of it for the scientific and biomedical domain, as well as the experiments and our final model for the BioMRC dataset (Pappas et al., 2020b). Finally, we are going to discuss an experiment we conducted with human evaluators, who answered a tiny subset of the dataset, in order to estimate human performance.

3.1 BERT-SciBERT overview

BERT is a state-of-the-art model that uses multilayer bi-directional Transformers (Vaswani et al., 2017) in order to solve various NLP tasks. The Transformers that the BERT model uses read the entire sequence at once and learn the contextual relations of the words using attention (Bahdanau et al., 2015) and fully-connected neural networks. In addition, positional embeddings are used, so that the order of the words is accounted for. The vocabulary of the model is given in WordPiece (Wu et al., 2016) tokens, which are very similar to Byte-Pair Encodings (BPEs) (Sennrich et al., 2016) and as a result no words are unknown for the model. A special [CLS] token is given in the start of the input text, which will contain the representation of the whole text after the training of the model. The model is trained in two stages, the pre-training stage, which is unsupervised learning, during which the model learns various aspects of the language, and the fine-tuning stage, where the model is fine-tuned to solve a specific NLP task such as Question Answering.

In the pre-training stage, BERT solves a special version of a language model task, which is called Masked Language Model, as well as the task of Next Sentence Prediction (Devlin et al., 2019). In the first task, a sentence has the 15% of its words replaced with a special

[MASK] token and the model then tries to predict the original words. The second task gets a pair of sentences, separated by a special [SEP] token and tries to predict whether the second sentence indeed followed the first one in the original corpus, or if it is a randomly selected sentence of the corpus. The model is jointly pre-trained for the two tasks and as both of them are unsupervised, the BERT model can be trained in the raw text of the corpora. The text that was used for the training was from the BookCorpus (Zhu et al., 2015) and the English Wikipedia.

In the fine-tuning stage, to solve a specific task, an output layer (which can be simple or complex) can be stacked over the BERT model and its parameters are trained, whilst the parameters of BERT are fine-tuned, in an end-to-end architecture. For example, in a Sentiment Analysis Task, a Multi Layer Perceptron (MLP) could take as input the [CLS] token, which contains the representation of the whole text and classify the sentence as Negative, Neutral or Positive.

SciBERT (Beltagy et al., 2019) is a BERT model, specifically trained on 1.14 million scientific articles from Semantic Scholar¹. From all these articles, 935k are in the biomedical domain, while the rest are computer science related. SciBERT has its own WordPiece (Wu et al., 2016) vocabulary using the SentencePiece library², which is better suited for tasks in the biomedical domain and outperformed BERT in BC5CDR (Li et al., 2016), ChemProt (Taboureau et al., 2011) and EBM-NLP (Nye et al., 2018) tasks. In our experiments we use SciBERT instead of BERT, as the vocabulary and articles of the BioMRC task is in the biomedical domain and the SciBERT model can benefit from that and yield better results than the vanilla BERT.

3.2 BERT problems with max sequence size

One major limitation of BERT is that the max sequence size of a given input is 512 tokens. This is a problem especially in the BioMRC task, as most of the abstracts are more than 512 tokens after the WordPiece (Wu et al., 2016) tokenization process.

The proposed solution is to split each abstract into sentences using a tool like NLTK (Bird et al., 2009) and use each sentence as an input to the SciBERT model. In this approach however, a way to combine each sentence representations is needed, so that we have a final abstract representation.

¹<https://www.semanticscholar.org/>

²<https://github.com/google/sentencepiece>

3.3 SciBERT as Embedding for ASReader - AOARReader

A first attempt to use SciBERT in order to solve the BioMRC task was to replace the GRUs (Cho et al., 2014) in the ASReader (Kadlec et al., 2016) and AOARReader (Cui et al., 2017) models with SciBERT in order to get the representation of the abstract and the title, leaving the rest architecture of the models intact. The aforementioned problem was immediately apparent, as the abstract was more than 512 tokens, so we first tried to truncate it to 512 tokens and later tried splitting the abstract into sentences and averaging the sentence representations.

A similar approach was to replace only the embedding layer of the ASReader and AOARReader models and instead use the embeddings from SciBERT which then will be given as input to the GRUs. However, with both approaches we obtained very poor results, regardless of the hyperparameters we used, so we had to discard them and come up with a better approach.

3.4 SciBERT models

As stated in the previous chapter, the task of BioMRC is to read the abstract and the title of an instance and to choose which candidate entity in the abstract fits in the 'XXXX' placeholder in the title. Since we need to split the abstract into sentences, we have a representation for each token for each sentence of the abstract from SciBERT. We also have to replace the 'XXXX' placeholder in the title with a [MASK] token, so that the SciBERT model outputs the representation containing the knowledge of the token which was hidden in the placeholder. The advantage of this approach is that the SciBERT model is already pre-trained for the Masked LM task, which utilizes the [MASK] token in the same way.

The idea is to take every token, that was originally in the abstract text an entity (in the form @entityN), and the [MASK] token from the title and assign a score to that token, which indicates how good that entity fits into the [MASK] token. Using this idea we experimented with various architectures in order to combine the representations of each entity with the [MASK] token and achieve high scores. More particularly, the experiments are discussed in the following subsections.

3.4.1 Multiplication, BiLinear multiplication with Logistic Regression

The first experiment was to element-wise multiply the embeddings of the entity and the mask and then use a simple logistic regression layer to output a score from 0 to 1. The second was to use a matrix with learnable parameters between the two representations and perform a BiLinear multiplication. Similarly to the first experiment, a simple logistic regression layer was used in order to get the score. Using this scheme we had a score for each entity in the abstract. The entity that the model yielded as an output was the entity with the largest score.

The reason that multiplication was used instead of summing the representations is that with the multiplication operator, both signals contribute to the output. This happens, because if one of the representations has values that are close to zero, then by multiplying it with the other representation or by using the BiLinear multiplication method, the output will also have values close to zero. As a result, both representations are important for the output and the model is forced not to be dependent on a single representation (e.g. the representation of the [MASK] token).

We trained the above models for the BioMRC task, but unfortunately the results were very poor in terms of accuracy, even compared to our baselines. Moreover, we observed that the results of the BiLinear multiplication were better than those of the simple element-wise multiplication. The problem was that, even though the multiplication operator is better than the addition, we still force the model to learn to modify the representations that the SciBERT model produces in order to solve the task. This is also apparent, since the BiLinear model, which has learnable parameters, performed better than the multiplication model, as it forced less the SciBERT model to modify its parameters and therefore did not have a big influence on the representations that it gave as an output. Thus, we discarded both ideas and searched for alternative approaches for the task.

3.4.2 SciBERT and Multi Layer Perceptron (MLP)

Using the above observations, we strived towards an approach where the model could use the representations from the SciBERT model without changing them. Instead, we tried to let the model learn how to combine the two representations in order to achieve a better score.

The way that we implemented this, is by concatenating the two representations and then passing them through a Multi Layer Perceptron (MLP) Layer that gives, as output, a single score. In this way, the model learns how to handle the concatenated representations

using the parameters of the MLP in order to score how well the entity fits the [MASK] placeholder. An additional factor which was considered, was that we keep the parameters of the SciBERT model frozen, ensuring that the model does not modify them and thus not affecting how the entity and [MASK] representations are given to the MLP. This is crucial, as it has the result of having a very small number of learnable parameters, making our model simple and easy to train, even for small research groups, with limited computational resources.

The implementation that was presented in detail in the previous paragraph, provided the mechanism to score how well each entity fits the [MASK] placeholder for every sentence in the abstract. However, we had to choose a strategy in order to get a single answer for the task. For this, we followed two approaches, the first, SciBERT-Sum-Reader (Pappas et al., 2020b), was to aggregate the scores for each unique entity by summing them, while the second, SciBERT-Max-Reader (Pappas et al., 2020b), was to choose the entity with the max score.

Both approaches achieved very good results, exceeding all baselines, as well as the AS-Reader and AOARReader models. The SciBERT-Max-Reader provided better results than the SciBERT-Sum-Reader, which is expected given the fact that by summing many low scores of a very frequent entity, the system might choose that instead of a not so frequent high scoring one.

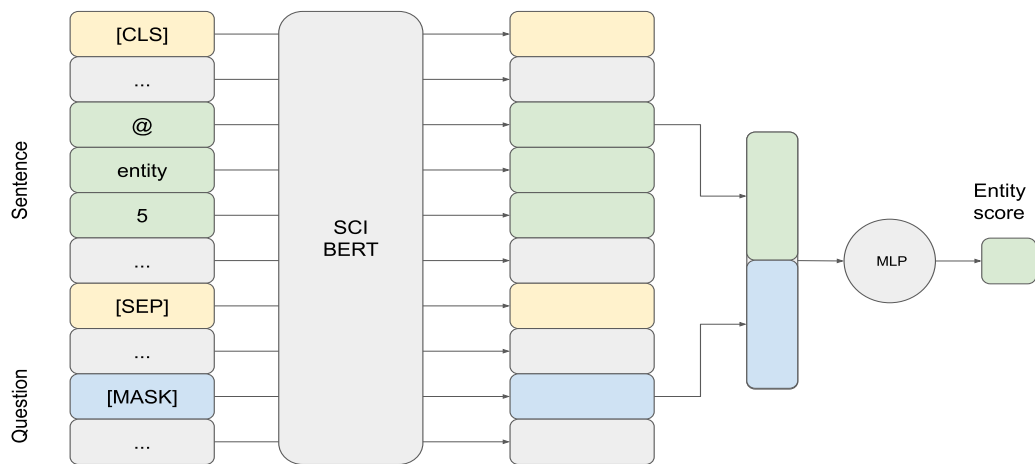


Fig. 3.1: Illustration of our SciBERT-based models. Each sentence of the passage is concatenated with the question and fed to SciBERT. The top-level embedding produced by SciBERT for the first sub-token of each candidate answer is concatenated with the top-level embedding of [MASK] (which replaces the placeholder) of the question, and they are fed to an MLP, which produces the score of the candidate answer. In SciBERT-Sum-Reader, the scores of multiple occurrences of the same candidate are summed, whereas SciBERT-Max-Reader takes their maximum.

3.5 Results of SciBERT-Reader models

The results of all the SciBERT-Reader models, compared to all baselines and the ASReader and AOAREader models can be seen in the following tables. In the tables, we show the accuracy scores of all models in the Train-Dev-Test sets of the BioMRC Lite dataset. The models were not tested on the Large BioMRC dataset, due to our limited computational resources. In addition, we show the training time and all trainable parameters of each method, as well as the parameters that originated from the Word Embeddings and from the Entity Embeddings respectively.

Method	BIOMRC Lite – Setting A						
	Train Acc	Dev Acc	Test Acc	Train Time	All Params	Word Embeds	Entity Embeds
BASE1	37.58	36.38	37.63	0	0	0	0
BASE2	22.50	23.10	21.73	0	0	0	0
BASE3	10.03	10.02	10.53	0	0	0	0
BASE3+	44.05	43.28	44.29	0	0	0	0
BASE4	56.48	57.36	56.50	0	0	0	0
ASREADER	84.63	62.29	62.38	18 x 0.92 hr	12.87M	12.69M	1.59M
AOAREADER	82.51	70.00	69.87	29 x 2.10 hr	12.87M	12.69M	1.59M
SCIBERT-SUM-READER	71.74	71.73	71.28	11 x 4.38 hr	154k	0	0
SCIBERT-MAX-READER	81.38	80.06	79.97	19 x 4.38 hr	154k	0	0

Tab. 3.1: Training, development, test accuracy (%) on BioMRC Lite in Setting A (global scope of entity identifiers), training times (epochs × time per epoch), and number of trainable parameters (total, word embedding parameters, entity identifier embedding parameters).

Method	BIOMRC Lite – Setting B						
	Train Acc	Dev Acc	Test Acc	Train Time	All Params	Word Embeds	Entity Embeds
BASE1	37.58	36.38	37.63	0	0	0	0
BASE2	22.50	23.10	21.73	0	0	0	0
BASE3	10.03	10.02	10.53	0	0	0	0
BASE3+	44.05	43.28	44.29	0	0	0	0
BASE4	56.48	57.36	56.50	0	0	0	0
ASREADER	79.64	66.19	66.19	18 x 0.65 hr	6.82M	6.66M	0.60k
AOAREADER	84.62	71.63	71.57	36 x 1.82 hr	6.82M	6.66M	0.60k
SCIBERT-SUM-READER	68.92	68.64	68.24	6 x 4.38 hr	154k	0	0
SCIBERT-MAX-READER	81.43	80.21	79.10	15 x 4.38 hr	154k	0	0

Tab. 3.2: Training, development, test accuracy (%) on BioMRC Lite in Setting B (local scope), training times (epochs × time per epoch), and number of trainable parameters (total, word embedding parameters, entity identifier embedding parameters).

For all the results in the above table, one-tailed significance tests were executed, using the Approximate Randomization (Dror et al., 2018) method, randomly swapping the answers to 50% of the questions for 10k iterations. In all the significance tests, the p-value was below 0.02, which indicates that all our results are statistically significant at a 0.05 significance level.

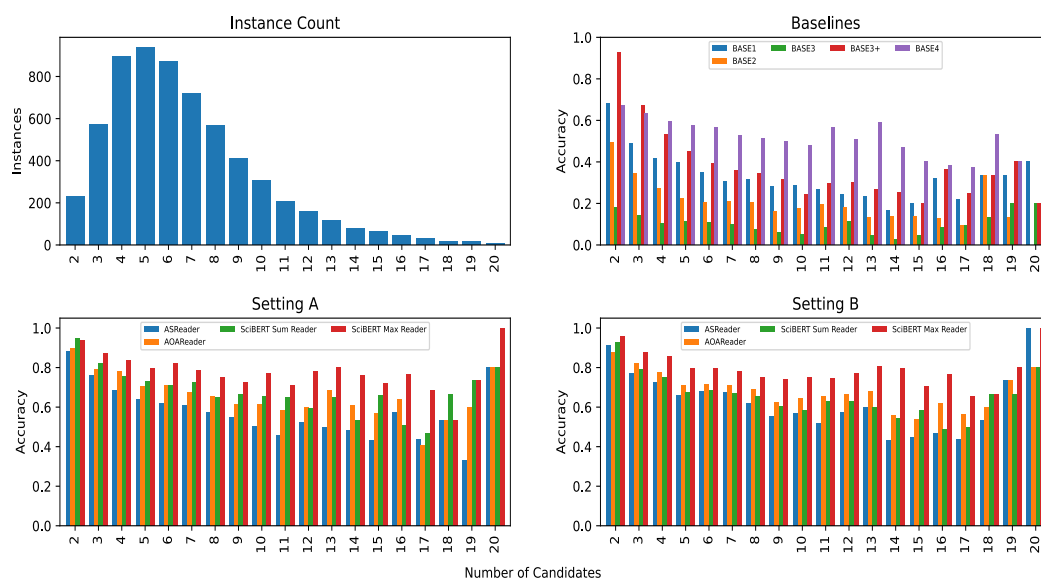


Fig. 3.2: More detailed statistics and results on the development subset of BioMRC Lite. Number of passage-question instances with 2, 3, ..., 20 candidate answers (top left). Accuracy (%) of the basic baselines (top right). Accuracy (%) of the neural models in Settings A (bottom left) and B (bottom right).

3.6 Human evaluation and MRC models comparison

In order to assess the cleanliness and quality of the new BioMRC dataset, we asked from some expert and non-expert human evaluators to try to answer a tiny subset of the dataset, following the steps of Pappas et al (Pappas et al., 2018). This tiny subset consisted of instances from both the Setting A and Setting B versions of the testing subset of the BioMRC dataset.

An HTML file was created for each instance, in which the entities in the abstract were hyperlinks that the human evaluator could choose. A choice indicates that the human evaluator thinks that this entity should be the answer, in the sense that it is the entity that was obscured behind the 'XXXX' placeholder. In Setting A, the entities in the HTML files that in the BioMRC dataset were in the format @entityN, were replaced by the corresponding word terms of the entity, so that the human evaluators could use their prior knowledge of the terms, in order to provide an answer for the instance. On the other hand, in Setting B, the entities remained as they are in the dataset, so that the human evaluators cannot use their prior knowledge to give an answer. This was done purposely, so that we can evaluate how the accuracy of the experts and non-experts changes in the two settings. Because the experts possess more experience and prior knowledge than the non-experts, we expect that they will perform better in Setting A, whereas in Setting B the difference in accuracy between the two will shrink.

Context:

OBJECTIVE: To assess the effects of socioeconomic factors on the association between parity and long-term maternal mortality. METHODS: This was a population-based cohort study of mothers with births registered in the Medical Birth Registry of Norway during the period 1967-2009. We estimated age-specific (40-69 years) cardiovascular and noncardiovascular mortality ratios by number of births using Cox proportional hazard models. To assess effect modification by mothers' attained education, we stratified on low (less than 11 years) and high (11 years or greater) educational level. We further evaluated fathers' mortality by number of births using the same analytical approach. RESULTS: Mothers with low education had higher mortality (cardiovascular: hazard ratio 2.62, 95% confidence interval [CI] 2.34-2.93, noncardiovascular: hazard ratio 1.67, 95% CI 1.62-1.73). Among mothers with low education, [cardiovascular mortality] increased linearly with each additional birth above one (P trend=.02). In contrast, among mothers with high education, [cardiovascular mortality] declined with added births (P trend=.045). For noncardiovascular mortality there was no association among mothers with low education, whereas mortality declined with increasing number of births among mothers with high education (P trend<.01). Father's mortality showed similar associations with number of births when stratified on maternal education. CONCLUSION: [women', 'Women']'s long-term mortality rose with number of births only for cardiovascular causes of [death] and only among mothers with low education. Partners of [women', 'Women'] with low education had similar increasing risk with increasing number of births. Maternal educational level is a strong modifier of the association between parity and long-term mortality. LEVEL OF EVIDENCE: II.

Question:

Association of XXXX's Reproductive History With Long-term Mortality and Effect of Socioeconomic Factors.

Your Answer:

Could answer the question using the context and prior knowledge
 @entity65 :: [cardiovascular mortality]

I could not answer the question
 I could answer the question with prior knowledge only
 I could answer the question using the context and prior knowledge
 I could answer the question using the context only

[Click me to save to file](#)

Fig. 3.3: HTML sample from the Setting A BioMRC tiny dataset for the human evaluation. The human evaluator picks one entity and it is highlighted across the document (with red color). Then the evaluator chooses whether he/she could answer the question using prior knowledge, the context, or both and submits his/her answer.

For the human evaluation, three non-experts and two experts contributed and answered 30 instances from Setting A and 30 (different) instances from Setting B. We also calculated the inter-annotator agreement, which is higher in the experts than in the non-experts. This is expected, as the experts all have a medical background and answer mainly from prior-knowledge, whereas the non-experts answer mainly from the context each instance. This is also visible in the table below.

Annotators (Setting)	Kappa
Experts (A)	70.23
Non Experts (A)	65.61
Experts (B)	72.30
Non Experts (B)	47.22

Tab. 3.3: Human agreement (Cohen's Kappa, %) on BioMRC tiny. Avg. pairwise scores for non-experts.

The results of the human evaluators in the tiny dataset, along with the corresponding results of the models are presented in the table below. We observe that the SciBERT-Max-Reader model surpassed both the experts and non-experts on Setting A, whereas in Setting B it surpassed the non-experts but not the experts. Of course, this does not prove that SciBERT-Max-Reader is better than the expert human evaluators, as there were very few instances per Setting.

Method	Setting A	Setting B
Experts (Avg)	85.00	61.67
Non-Experts (Avg)	81.67	55.56
ASREADER	66.67	46.67
AOAREADER	70.00	56.67
SCIBERT-SUM-READER	70.00	56.67
SCIBERT-MAX-READER	90.00	60.00

Tab. 3.4: Accuracy (%) on BioMRC tiny. Best human and model scores shown in bold.

Transfer learning to BioASQ

In this chapter we are going to explore the Question Answering task, BioASQ Task 8B Phase B, which is related to BioMRC (Pappas et al., 2020b), since it also resides on the biomedical domain. The task is part of the BioASQ 8 challenge (Tsatsaronis et al., 2015). The questions are factoids (requiring a single entity as answer) or list-questions (requiring a list of entities each).

More particularly, the BioASQ Task 8B challenge consists of two phases. In the first phase, questions are given and the participants are required to implement models that return relevant biomedical articles and snippets which contain the answers to the questions. In the second phase, the gold snippets are already provided by the challenge and the participants are asked to create models that extract an exact answer (for factoid questions), or a list of exact answers (for list questions) to the question. The participants must provide a list of a maximum of 5 exact answers for the factoid questions and a list of a maximum of 10 exact answers for the list questions. A subset of the questions in the second phase are "yes/no" and summary questions, which are treated separately, but in this thesis we will not consider these questions at all.

4.1 Evaluation Measures of BioASQ Task 8B Phase B

Each part of the BioASQ Task 8B Phase B task uses different evaluation measures. For the factoid questions, three evaluation measures are used. The first two are accuracy evaluation measures and are referred as Lenient and Strict accuracy respectively, whereas the third evaluation measure is the Mean Reciprocal Rank (MRR) (Craswell, 2009). Both Lenient and Strict accuracies are calculated as the mean of the questions in which the model provided a correct answer. Given a list of model answers for a question, if the lowercase text of any model answer, in Lenient accuracy, or the first model answer, in Strict accuracy, matches the lowercase text of the gold answer, then the question is considered to have been answered correctly. The MRR is calculated as the mean of a score for each question. A question's score is the fraction of 1 by the index position of the model answers list, in which the lowercase text matches the lowercase text of the gold answer. This position index starts from 1 and if none of the model answers match the gold answer, then the score for that question is 0.

For the list questions, precision recall and f1-score evaluation measures are used. Similarly to the factoid questions, a model answer is considered correct, if the lowercase text of the model answer matches exactly the lowercase of the gold answer.

The BioASQ challenge provided code written in Java for the participants to evaluate their models. However, it proved to be time-consuming to run it every time separately from our Python code and debugging was not possible. For these reasons, we implemented all the evaluation measures in Python and tested that they produce the same results.

4.2 Zero-shot using SciBERT-Reader models

In this section we are going to discuss our first attempt at solving the aforementioned task. More specifically, we experimented with a zero-shot learning technique, which refers to the process of a model, which has been trained for a task, to attempt to solve a different task that it has not been trained on, without additional training or fine-tuning.

We are going to use the pre-trained SciBERT (Beltagy et al., 2019) models which we introduced in chapter 3, since the BioMRC task, which lies in the biomedical domain, is very close contextually to the BioASQ challenge. In addition, both tasks are Question Answering tasks, albeit the latter is not cloze-type (Taylor, 1953) and therefore is more complex. For that reason, we do not expect extremely good results on the task, but rather investigate whether the pre-trained SciBERT model on the BioMRC task gives better results than a randomly initialized SciBERT model.

4.2.1 Pre-trained SciBERT-Reader model approach

Since the SciBERT-Reader model was built for the BioMRC task, it cannot be directly used for the BioASQ task. Since BioMRC was a cloze-type Question Answering task, the SciBERT-Reader models combined each candidate entity in the form '@entityN' with the 'XXXX' placeholder entity and passed it through an MLP layer in order to score each candidate entity and choose the final answer using these scores. However, BioASQ is not a cloze-type Question Answering task and therefore are no candidate entities, or equivalently, every n-gram in the gold snippets is a candidate entity. Furthermore, there is not any placeholder entity like the 'XXXX' entity in BioMRC, thus the scoring mechanism cannot work exactly as in the task of BioMRC. We bypass this shortcoming by manually adding a placeholder entity in the end of the question. We also experimented by adding the placeholder at the start of the question but obtained inferior results.

Because taking into account each n-gram of the snippets as a candidate entity is a very costly approach, a solution was to treat each word of the snippets as a candidate, scoring

each one and then implementing a post-processing function to find and rank the best n-grams using these scores. Moreover, we could not concatenate all the snippets into a large snippet, because of the max sequence length limit of the SciBERT model. For that reason, we pass each snippet combined with the question separately from SciBERT to get the representations needed for the MLP to calculate the score for each word.

The post-processing function to find and rank the best n-grams is done separately for each snippet as well, providing a single exact answer per snippet. The idea is to find the sequence of words with the largest sum (or mean) score in a snippet. At first, all stopwords (from NLTK (Bird et al., 2009)) and words that are not a noun, verb or adjective (using `spacy`¹), are assigned a score of -999.0, so that they are not included in an exact answer. Then, min-max normalization of the word scores is performed twice, once to bring all scores to the [0, 1] range and then to all the non-zero scores, so that the higher scores are boosted. By inspecting the previous test subsets of the BioASQ challenge, as well as the training subset of the current challenge, we discovered that the average length of a factoid instance answer is a trigram, while the average length of a list instance answer is a bigram. Due to this factor, the highest scoring ngrams of length < 3 for factoid and < 2 for lists are chosen as the exact answers for each snippet. Finally, the best ngrams are ranked from the highest to the lowest score and the first 5 are given as the final answer to the task.

One technique we experimented with is to group the ngrams from each snippet using the `fuzzywuzzy`² module, which uses edit distance in order to compute the string matching ratio of the two texts. Using this string matching ratio, we grouped the answers which were too similar (over 90%) and allowed only the ngram with the highest score out of each group to be selected for the final answer. This allowed the model to output a larger variety of exact answers and not repeat the same multiple times.

A major problem with this approach is that the BioMRC task has entities in the form `@entityN` in its text and therefore the pre-trained model that we use expects these tokens in the input. The BioASQ task however does not use this notation, so all the knowledge of the biomedical entities that the SciBERT model had been trained on, is not being used. In order to fix this issue we used the mappings from the Pubtator tool (Wei et al., 2012a) in the snippets of the BioASQ instances, performing the same process in order to get the biomedical entities in the correct form. In this way, the entities are scored correctly from the pre-trained SciBERT model. The entities are then mapped back to the original text and the score of each entity is assigned to every word that they contain. Thus, this annotation process affected only the scoring process and not the post-processing function, which remains the same as described above.

¹<https://github.com/explosion/spaCy>

²<https://github.com/seatgeek/fuzzywuzzy>

Besides the zero-shot approach, we also tried to fine-tune the parameters of the SciBERT model, by training them on the train set of the BioASQ dataset. The training instances were created from each question and its snippets by finding the exact answer or answers in the snippets, using string matching and creating a mask, with length equal to the length of all the snippets concatenated. The mask had a value of '0' if a word was not in the answers and a value of '1' if it was, thus creating a target vector for the training. A sigmoid layer was applied to all the word scores and the Binary Cross Entropy (BCE) loss is calculated for each word. The loss is then calculated as the mean of the BCE loss of all the words. For the evaluation and the early stopping we used the MRR evaluation measure. The best epoch in our validation set was then used for the test set of the BioASQ task.

Even after immense testing and experimenting with hyperparameters for the above processes, the results were very poor, especially compared to the competition for the BioASQ challenge. This could be an indication that the BioMRC task is not related to the BioASQ task, as we firstly thought. The fine-tuning on the BioASQ train set gave a small boost to the results, but it was inconsistent and still could not compete with the competition.

4.3 SpanBERT, BioBERT approaches

The above results led us to experiment with other versions of BERT models apart from SciBERT like SpanBERT (Joshi et al., 2020) and BioBERT (Lee et al., 2019). We tried using these models as encoders instead of SciBERT inside the architecture described above. In addition, we experimented using the architecture for Question Answering implemented by the Huggingface team³. The latter outputs directly the start and end span for the exact answer in each snippet. One important note is that because these models have not seen the biomedical entities in the @entityN format, like the pre-trained SciBERT-Reader model, the whole annotation process described in the previous section has been stripped out from the architecture pipeline.

SpanBERT is a version of BERT (Devlin et al., 2019), which is specifically trained to create better representations and predictions of spans of text. It is based on a contiguous span masking process and introduced the span-boundary objective (SBO), for which the model predicts an entire masked span based on the boundary observed tokens. This helps SpanBERT to achieve a better performance at many Question Answering tasks such as SQUAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), TriviaQA (Joshi et al., 2017), SearchQA, HotpotQA, and Natural Questions (Kwiatkowski et al., 2019) compared to BERT.

³<https://github.com/huggingface>

The first attempt to use SpanBERT was to just use it as an encoder, in order to get the word and the [MASK] representations in the same architecture as described in section 4.2. This attempt was not successful as it achieved poor results, even after fine-tuning the model in the BioASQ training set. The second attempt was to use the pre-built architecture for Question Answering, as implemented by the Huggingface team. This architecture gave as an output the start and the end of the spans for the exact answers per snippet, along with their corresponding scores, which were used to create candidate ngrams as in the case of the SciBERT model. These exact answers were subsequently grouped and ranked using the post-processing function which was analyzed in the previous section.

The results in this method were slightly better than using SpanBERT as an encoder, but still no great improvements were achieved. Also, SpanBERT has not been pre-trained on any corpora which include biomedical concepts and entities. This could be an indication, that although it has been trained to be good for the purposes of predicting spans of text, it cannot perform well for biomedical questions.

BioBERT is a BERT model that was pre-trained in the same way as BERT, but also included citations and abstracts from PubMed and full-text journal articles from PMC (PubMed Central)⁴ in its training corpora. BioBERT was also fine-tuned for the Named Entity Recognition, Relation Extraction and Question Answering tasks, on a variety of biomedical datasets, which include the BioASQ dataset. BioBERT performed better on most of these tasks compared to the base BERT model and to the, at that time, state-of-the-art approaches for each task. BioBERT was also used as a component of the architecture of a research team in the BioASQ 7 challenge, which obtained high scores.

However, the pre-trained BioBERT model was implemented and trained using TensorFlow⁵, which is not compatible with our software, since it is implemented in Pytorch⁶. Therefore, we experimented by porting the BioBERT model in Pytorch, using the Huggingface environment and loading the pre-trained parameters from the TensorFlow model into our implementation. We then used the BioBERT model in the same way and with the same post-processing, as we did with the SpanBERT model. However, the results were very discouraging, which is probably due to the fact that the porting process was not flawless. That indicated that additional effort and tinkering with the porting mechanism must be provided. For this reason, we discarded the model and moved on to better approaches.

⁴<https://www.ncbi.nlm.nih.gov/pmc/>

⁵<https://www.tensorflow.org/>

⁶<https://pytorch.org/>

4.4 Text-to-Text Transfer Transformer (T5) approach

A recent architecture, which is not specific to the biomedical domain, but rather tries to encapsulate a more universal representation of knowledge is the Text-to-Text Transfer Transformer (T5) (Raffel et al., 2020). As the name suggests, this is a Transformer-based model (Vaswani et al., 2017), which is trained on many NLP tasks both unsupervised (e.g. Masked Language Modelling) and supervised (e.g. Machine Translation, Sentence Entailment, Summarization, Text Classification, Question Answering, etc). More particularly, it is a multi-task sequence-to-sequence transfer-learning model, which needs the input to be formatted in a specific way in order to perform each separate task. For the task of Question Answering, a text is given with the question and the context and the model responds with a text containing the answer. The model was evaluated on many datasets and for many different tasks, including the SQUAD dataset, which produced very good results. T5 is a generative model, which means that, in contrast to the models that we have seen until this point, it does not score the words of the snippets or provide start and end spans for the exact answers, but it rather uses the snippets as a vocabulary of words, which are available in order to generate exact answers.

In order to use the T5 model for the BioASQ task, we used the above formulation, generating one or more exact answers for each snippet of an instance. However, the model does not score the exact answers it generates. We therefore solved this issue, by computing the score using the mean of the logits from the indices belonging to the generated tokens. This requires a second pass through the model, though, which introduces additional computation time. Moreover, having scores for each generated exact answer, allows to use the same post-processing for grouping and ranking, as in the previous sections. The training process of the T5 model is not very different than that of the previous approaches. The only difference is that in the case of multiple exact answers (e.g. in the list questions), the target answers must be given to the model as a text sequence, separated either by spaces or a special delimiter. The results from the T5 model were significantly better than the previous approaches, which is quite surprising, given that it is a universal model and it is not specifically trained for the biomedical domain. One explanation of the better performance of the T5 model, could be that it has been trained on much larger corpora compared to the previous models that we tested.

In order to further enhance the results of the model, we tried fine-tuning the model on the SQUAD and BioMRC datasets. The SQUAD dataset, could be used directly without any changes, as it has the same format as the BioASQ dataset, albeit having only factoid questions. The BioMRC dataset however is in a different format and all the biomedical entities are in the form of @entityN, on which the T5 model has not been trained on. For

this reason, we replaced all the entities with their corresponding text, which is available in the BioMRC dataset.

The results on the dev set of BioASQ were mixed and inconsistent, but that was expected, since the datasets were not that close to BioASQ. However, they could be used as a pre-training step, so that the model could use the contextual knowledge from the datasets and learn to solve the task better by fine-tuning on the train set of the BioASQ dataset.

By experimenting using the T5 model, we discovered that the post-processing function, which groups and ranks the generated answers was not beneficial at all. This was contrary to the previous models, for which the post-processing was helpful and achieved slightly better results. In fact, by removing it, the results improved vastly in all of our experiments.

As we discussed in the previous chapter, where we analyzed the problems of the BERT-based models, the max sequence length limitation prevented the model to read all the snippets at once. This means that the model could not use the contextual knowledge of all the snippets at once and was forced to give at least one exact answer per snippet, even in the case that the snippet does not include a span of the correct answer. By investigating the code of the T5 model, we found that the max sequence limit is not present and the input sequence can be any size, provided that the memory of the hardware can support it. Even though there is no limit, problems could still persist, as the T5 model was not specifically trained for long input sequences. Thus, more experimentation is required in order to obtain a clearer insight.

For this reason, we implemented a full context approach using the T5 model, where all the snippets are concatenated into a single text and then are given as input to the model in the correct format for Question Answering. The generation of the exact answers is also modified, as we have only one large concatenated text as input and thus one generated exact answer, instead of one exact answer per snippet. The Huggingface implementation of the T5 model, enables us to generate many exact answers using beam search and by applying other parameters, we can control the variety of the generated answers. The latter is performed by penalizing very similar generated text and it is crucial to the task, as most of the evaluation measures used in BioASQ favor a list of different exact answers for a question instead of a list with the same or similar answers. This is trivial, especially in the list question instances, where it is required to list all the exact answers instead of just one. We have performed some basic experiments using the full context T5 model, which achieved very good results. This fact could indicate that the context knowledge across all snippets is very beneficial for the task, but more testing and experimenting is required.

4.5 BioASQ in BioMRC format

Previously, we stated that we used BioMRC instances as a pre-training stage for some of our experiments. However, these instances are not in the same format as the BioASQ instances, which can confuse the model. Since the BioMRC dataset has a large amount of instances, the idea of modifying the instances and bring them into the same format of BioASQ, would require recruiting many human annotators, which is very costly. For that reason, we decided to do the reverse process and create a new BioASQ task, in which all the questions are transformed into cloze-type style. This is manageable, due to BioASQ's small amount of instances, but of course does not solve the original task. Instead, it could provide us with some indication, on whether the BioMRC dataset can be beneficial as a pre-training step for another biomedical Question Answering task.

In order to transform the BioASQ instances into the format of BioMRC, we applied a combination of regular expressions and human annotation. To achieve this, we got the questions of the BioASQ factoid instances and replaced the wh-words ("which", "when", "where", "what", etc) with 'XXXX' tokens. Then we inverted the sentence so that it is in the statement form instead of the question form and also changed the conjunction and the tense of the sentence's verbs when it was required. We performed this human annotation for all 941 factoid questions of BioASQ, creating the BioASQ MOD dataset.

We then performed all of our experiments on the modified BioASQ dataset. The results were more or less the same as in the original dataset, which could either indicate that the process did not help the model, or that the relative low scores in the task reside in other factors. Still, the BioASQ MOD dataset is a solid contribution and can be used by researchers to experiment with their models in both cloze-type and normal questions, in the same domain and context.

4.6 Results and problems of BioASQ

In this section we are going to report some of the models' results in the BioASQ dataset and additionally provide an analysis of our findings. Furthermore, as we have already mentioned in many sections of this thesis, we encountered many issues with the BioASQ dataset, concerning both the evaluation measures and the gold answers of the questions.

In the results we compare the performance scores of the T5 model on different training datasets. The datasets which are used in the training process are the SQUAD dataset, the BioMRC dataset and the BioASQ dataset. Both SQUAD and BioMRC contain a large number of instances. Training the T5 model in a different task than the one that is evaluated on, despite the similarities, cannot guarantee good results. For that reason, we used three

Method	Factoid			List		
	Strict Acc	Lenient Acc	MRR	Precision	Recall	F1
T5 - No Training	13.39%	14.29%	13.84%	5.62%	2.93%	3.83%
T5 - BioASQ Training	15.18%	15.18%	15.18%	8.77%	3.91%	5.30%
T5 - SQUAD 1K Training	14.29%	15.18%	14.73%	4.89%	2.39%	3.21%
T5 - BioMRC 1K Training	16.07%	16.07%	16.07%	5.87%	1.85%	2.79%

Tab. 4.1: Results of the T5 model using the post-processing function and without using the custom scoring function.

Method	Factoid			List		
	Strict Acc	Lenient Acc	MRR	Precision	Recall	F1
T5 - No Training	19.64%	33.04%	24.05%	13.01%	12.90%	12.07%
T5 - BioASQ Training	23.21%	34.82%	27.19%	12.39%	13.45%	12.30%
T5 - SQUAD 1K Training	21.43%	38.39%	27.05%	12.72%	13.43%	12.30%
T5 - BioMRC 1K Training	19.64%	31.25%	24.29%	12.79%	10.08%	10.62%

Tab. 4.2: Results of the T5 model without using the post-processing function and without using the custom scoring function.

subsets of the datasets. For the SQUAD dataset we used 1K, 5K and 100K subsets, whereas for BioMRC we used 1K, 5K and 50K subsets. Also, we fine-tuned the models on the BioASQ task after pre-training on each subset of the datasets. In table 4.1 we show the scores of the T5 model using the post-processing function, while in table 4.2 we provide the scores of the T5 model without the post-processing function. Both tables show the scores without our custom scoring function for each candidate answer and use only the 1K version of the datasets for training. In table 4.3 we have provided the results for each individual experiment, with all versions of the datasets, including the modified version of BioASQ, in which we transformed all the instances in the format of BioMRC. All experiments do not use the post-processing function but apply the custom scoring function, which scores each candidate answer based on the logits of the model, as we discussed in section 4.4 .

We can clearly see by inspecting the results that the post-processing function had an impact on the model's results, since by removing it, all the experimental results were improved. Similarly, we observe, that by using the custom function to score the candidate answers (ngrams) that each T5 model generates, we get slightly better results. This is also the reason that most of our experiments were performed using this custom function. From table 4.3 we can clearly see that when the large subsets of the SQUAD and BioMRC datasets were used, the results deteriorated. This is expected, as the model learns to solve a similar task in a different domain, in the case of SQUAD, and in a similar domain but with a slightly different task in the case of BioMRC. Additionally, by fine-tuning the models of these experiments to the BioASQ challenge, we observe some improvements, especially in the large subsets, which further supports our argument.

Method	Factoid			List		
	Strict Acc	Lenient Acc	MRR	Precision	Recall	F1
T5 - No Training	18.75%	33.04%	24.66%	13.88%	12.62%	12.56%
T5 - BioASQ Training	20.54%	39.29%	27.95%	16.16%	16.03%	14.98%
T5 - SQUAD 1K Training	21.43%	34.82%	25.86%	15.07%	14.82%	14.20%
T5 - SQUAD 5K Training	22.32%	36.61%	28.15%	11.30%	12.10%	11.00%
T5 - SQUAD 100K Training	12.50%	19.64%	15.51%	1.09%	0.43%	0.62%
T5 - SQUAD 1K + BioASQ	23.21%	36.61%	28.56%	14.31%	16.53%	14.71%
T5 - SQUAD 5K + BioASQ	22.32%	36.61%	28.15%	11.30%	12.10%	11.00%
T5 - SQUAD 100K + BioASQ	14.29%	28.57%	20.71%	7.72%	7.02%	6.74%
T5 - BioMRC 1K Training	21.43%	33.04%	25.88%	14.09%	13.11%	12.73%
T5 - BioMRC 5K Training	15.18%	27.68%	20.04%	13.22%	13.24%	12.80%
T5 - BioMRC 50K Training	8.93%	13.39%	10.67%	9.13%	6.63%	7.06%
T5 - BioMRC 1K + BioASQ	21.43%	33.04%	25.88%	14.09%	13.11%	12.73%
T5 - BioMRC 5K + BioASQ	24.11%	40.18%	30.64%	15.72%	15.23%	14.41%
T5 - BioMRC 50K + BioASQ	17.86%	29.46%	22.13%	16.34%	15.15%	14.45%
T5 - BioMRC 1K + BioASQ MOD	21.43%	33.04%	25.88%	14.09%	13.11%	12.73%
T5 - BioMRC 5K + BioASQ MOD	17.86%	32.14%	23.35%	15.54%	15.12%	14.35%
T5 - BioMRC 50K + BioASQ MOD	22.32%	30.36%	25.18%	14.06%	14.56%	13.18%

Tab. 4.3: Results of the T5 model without using the post-processing function and with our custom scoring function.

Another observation, is that in the majority of experiments, in which pre-training was applied on other datasets, we did not obtain better results than simply training on the BioASQ dataset. This was not the case for the BioMRC 5K subset, which when used as pre-training the model achieved better results for the factoid instances. Specifically, it scored 2.69% more in the MRR evaluation measure and 3.57% more in the Strict Acc than just training on the BioASQ dataset. However, the difference in the scores is small and thus more experimenting should be executed to prove that pre-training on BioMRC can achieve consistently better results on the BioASQ task.

Furthermore, the modified version of the BioASQ task achieved worse results when the model was pre-trained on the BioMRC dataset. This was unexpected, since we thought that reformatting the BioASQ task to match the format of the BioMRC would achieve better results. This could be an indication of the fact that the BioASQ task is not as similar as the BioMRC task and consequently their different format did not have a major effect on what the model learned.

The persistent low scores of our models made us question, whether the BioASQ task was too difficult, or if some other factors were also contributing. So, we performed an error analysis of the results of the model, debugging our implemented BioASQ evaluation measures. In this analysis, we inspected both predicted and gold answers of the instances, when the predicted answer should have been considered a valid answer, but falsely did not. In order to achieve this, we used the module from fuzzywuzzy, instead of a simple lowercase comparison, to compare the predicted answer with the gold one. If the similarity ratio was bigger or equal to 90%, we retrieved both the prediction and the gold answer for further inspection. Below are presented some examples of our findings.

Question	Prediction	Gold Answer	Problem
What is the function of LOX proteins in the ECM?	cross-linking collagens and elastin	the best-studied role of lox enzymes is the remodeling of the extracellular matrix (ecm) in animals by cross-linking collagens and elastin.	Long Answer
What is an exosome?	extracellular vesicles (evs)	exosomes are a subset of extracellular vesicles (evs) that have important roles in intercellular communication. they contain and carry bioactive molecules within their membranes which are delivered to target cells.	Long Answer
Which is the main protein in brown adipose tissue (BAT) active in thermogenesis?	uncoupling protein 1 (ucp1)	uncoupling protein 1	Same Answer
What is the 3D tomography imaging technique for diagnosis of eye disease?	optical coherence tomography	optical coherence tomography.	Same Answer
In what percentage of skeletal muscle fibers is dystrophin expression restored after PPMO- mediated exon skipping?	nearly 100%	100%	Better Answer
Where in the body would the navicular bone be found?	medial side of the foot	foot	Better Answer

Tab. 4.4: Problems of various question-answer instances from the BioASQ dataset. The predictions of the T5 model are also showed in the table. These predictions were incorrect using the BioASQ evaluation measures, but correct using our custom evaluation measures. The last column mentions the problem with each instance.

As we can see, the comparison function that is being used in the evaluation measures is clearly problematic. In many cases, the predicted answer is correct or even better than the suggested gold answer, but it is considered as incorrect. Furthermore, we observe that in some examples, the gold answers are too long and/or are not exact answers, but rather ideal answers or long explanations. This is contrary to the description of the BioASQ task about exact answers and ideal answers, as it specifies that the exact answers are mostly entities which answer directly the question.

These problems have an immediate effect on the results of the models, since they ignore correct answers. One could even say that the models on the top of the competition have overfitted on the dataset, as they have to predict answers that follow the dataset's bias and fine-details to achieve large scores on the evaluation measures. To prove our point, we used the aforementioned comparison function instead of the lowercase equality function in the implemented BioASQ evaluation measures and tested the T5 model. We report below the results of this process, in which one can observe that the scores have been increased dramatically. This could mean that the BioASQ organizers should reconsider the evaluation measures and functions for the competition, so that the models trained for the task get rewarded for better generalization and not for learning the fine details of the dataset.

Method	Factoid			List		
	Strict Acc	Lenient Acc	MRR	Precision	Recall	F1
T5 - BioASQ Evaluation Measures	18.75%	33.04%	24.66%	13.88%	12.62%	12.56%
T5 - Custom Evaluation Measures	42.86%	65.18%	52.70%	46.78%	39.86%	40.00%

Tab. 4.5: Results of T5 trained on BioASQ dataset using the original BioASQ evaluation measures and the custom evaluation measures using the fuzzywuzzy module.

Conclusions and future work

In this chapter, we are going to see an overview of the work that has been done as part of this MSc thesis. We are going to discuss what was achieved through this work and the impact that it has in the field of Machine Learning Comprehension (MRC) (Hermann et al., 2015) and in particular to the biomedical domain. Furthermore, we write about our contributions, which were not closely related to the main work of this thesis, but were performed in the same time frame of researching and experimenting. Lastly, we will refer to all the experiments and ideas that could not be explored in this thesis, but will be part of our future work.

5.1 Overview

In this thesis, we introduced BioMRC (Pappas et al., 2020b), a novel cloze-type MRC dataset for the biomedical domain, which was created using the abstracts of Pubmed Central¹. This dataset solved the problems of the previous work of the BioRead dataset (Pappas et al., 2018), by reducing the noise in the text and by using the Pubtator (Wei et al., 2012a) web tool for finding the biomedical entities, which created less noise and mistakes than MetaMap (Aronson and Lang, 2010). The ASReader (Kadlec et al., 2016) and AOARReader (Cui et al., 2017) models, which were already tested on the BioRead dataset, were implemented and tested on our dataset, in order to compare the results of the two datasets. Two new models were introduced as well, SciBERT-Sum-Reader and SciBERT-Max-Reader (Pappas et al., 2020b), which used SciBERT (Beltagy et al., 2019), a BERT (Devlin et al., 2019) model trained on 1.14 million scientific articles from Semantic Scholar².

The rest of our work concerned the solution of the BioASQ Task8B Phase B (Tsatsaronis et al., 2015), which is a Question Answering (QA) task in the biomedical domain, where a model answers a question using some snippets, as the context that contains the answer. We experimented solving the above task using a zero-shot learning approach, by using the SciBERT-Sum-Reader and SciBERT-Max-Reader models pre-trained on the BioMRC dataset, with a slightly modified architecture in order to work for the BioASQ Task, in which a pre-processing function and a post-processing function were added. The pre-processing function was used in order to incorporate the entity embeddings that were present in the

¹<https://www.ncbi.nlm.nih.gov/pmc/>

²<https://www.semanticscholar.org/>

BioMRC dataset and in the BioASQ questions and snippets. The post-processing function was used to convert the results of the models that were designed for cloze-type instances to the required format and then get n-gram exact answers as a result, ranking them based on their score, instead of a single entity answer which was the case in BioMRC. We also attempted to fine-tune the above models on the train set of the BioASQ Task 8B Phase B dataset. Furthermore, we used the same architecture incorporating other BERT-based models, namely SpanBERT (Joshi et al., 2020) and BioBERT (Lee et al., 2019), which were pre-trained on different datasets and tasks and could therefore benefit the results for our task. These models were also fine-tuned on the train set of the BioASQ Task 8B Phase B.

In our effort to try another approach for the BioASQ task, we used the Text-to-Text Transfer Transformer (T5) model (Raffel et al., 2020) to generate the exact answers. We experimented by training and fine-tuning the T5 model on combinations of the BioMRC, SQUAD (Rajpurkar et al., 2016) and BioASQ datasets, in order to find which one achieves the best results. We also implemented a function to score the exact answers generated by the T5 model and part of the aforementioned post-processing function to rank the answers, to get better results. Moreover, we converted manually all the factoid questions from the BioASQ dataset into a cloze-type format, similar to the BioMRC dataset, and performed experiments, in order to figure out whether that could help the T5 model pre-trained on the BioMRC dataset to perform better.

Lastly, we performed error analysis on the results of our best architecture for the task, which uses the T5 model. In this analysis we found that many of the errors were not justified, as they either matched the gold answer provided by BioASQ or the gold answer was not a correct exact answer. This led us to investigate the evaluation measures which were used in the BioASQ Task 8B Phase B and to suggest better evaluation measures using edit distance based methods by utilizing the `fuzzuwuzzy` python module³, which accepts model predictions as correct answers, even in the case that the lowercase text of the predictions does not match exactly the gold answer. For that matter, we also listed some examples of answers that were too long and too detailed to be considered as exact answers for a model to predict and which should be better considered as ideal answers.

5.2 Accomplishments

The accomplishments of our work were focused on the fields of MRC and Question Answering in the biomedical domain. BioMRC, our novel cloze-type MRC dataset, can be used by research teams, in order to develop and experiment models for cloze-type Question Answering. Additionally, it can be used as a pre-training step of a model that performs Question Answering tasks like SQUAD or BioASQ, providing a large number of good

³<https://github.com/seatgeek/fuzzywuzzy>

quality instances. This was not available in any previous work apart from the BioRead dataset, which this dataset relies on and attempts to improve. Along the dataset, two new models were introduced for the cloze-type Question Answering task, which performed very well in the dataset, surpassing the previous models by a good margin. These models, could also be used in other related tasks and provide a good indication that SciBERT and other Transformer-based (Vaswani et al., 2017) models are taking over the MRC field and consistently achieve better results than any other approach.

Moreover, by using the above novel models on the BioASQ task embedded in a zero-shot learning architecture, we concluded that despite our initial expectations that the BioMRC and BioASQ tasks were very close, that was not the case. However, we observed that both models, as well as the T5 model, when pre-trained on the BioMRC dataset and then trained on the BioASQ dataset, get a small boost in the result scores, when compared to only training on the BioASQ dataset. From these two observations, we can speculate that the BioMRC task can indeed help other MRC and Question Answering tasks, by using the proper architectures and models. This performance boost can of course vary, depending on the models used and the contextual compatibility of the tasks.

In addition, we manually converted all the BioASQ factoid questions to cloze-type questions, in an effort to boost even more the performance of the T5 model, when pre-trained on the BioMRC dataset. The intention was to match BioASQ to the format of the BioMRC task and thus to eliminate one of the barriers that could prevent an even bigger boost on the performance by pre-training on the BioMRC dataset. A by-product of the above process is that a new cloze-type factoid BioASQ dataset is now available, which could be used as a training set for automatically transforming biomedical questions to the cloze-type format and then one could use a model like the SciBERT-Sum-Reader or SciBERT-Max-Reader to solve it.

Lastly, with the analysis on the evaluation measures and the gold answers of the BioASQ Task 8B Phase B, new insights are provided. These insights can be used by the organizers of BioASQ to fix the errors of the problematic gold answers in the task, as well as, to improve the evaluation measures in order to include predictions that match contextually the correct answer, even if the text is slightly different.

5.3 Extra contributions

While running all the experiments and writing this thesis, some extra work has been done, which is not necessarily directly linked to the main work of the thesis. In this section, we are going to briefly discuss the extra contributions that were performed in that manner.

5.3.1 Covid-19 Search Engine

Due to the Covid-19 pandemic outbreak in the time of writing this thesis, the whole scientific research community has focused their effort into trying to provide help in any way possible. To this extent, we wanted to participate as well, so that we could help the scientific community to come up with a solution for the virus and at the same time help the masses to get quick and correct answers from peer-reviewed sources about issues and inquires that they may have.

For that reason, we constructed an online search engine⁴ based on the CORD-19 Open Research Dataset⁵, which consists of many papers that are related to the SARS-CoV-2 virus. From all these papers, the titles and sections are then indexed into an Elasticsearch⁶ search engine for snippet retrieval as presented in the work of Pappas et al. in BioASQ 7 (Pappas et al., 2020a).

The users of the search engine specify a question and optionally a section name that the model should retrieve its answers from and submit it. The results are then displayed as a list of sections from articles with a descending relevance score. If the model predicts that a sentence in the section or title text has a high probability of being a correct snippet that answers the question, then it assigns it a yellow color, so that the user can see it.

AUEB NLP Group COVID-19 Search Engine

An Experimental Document and Snippet Retrieval Search Engine for the [COVID-19 Open Research Dataset \(CORD-19\)](#) based on [award winning](#) technology developed by [AUEB's NLP Group](#) for the [BioASQ challenge](#)

WRITE YOUR QUESTION

Pregnant women and COVID-19

OR SELECT AN EXAMPLE QUESTION

Pregnant women and COVID-19

OPTIONALLY, SPECIFY SECTIONS YOU WANT TO SEARCH:

Narrow down your section.

OR SELECT A SECTION FROM THIS LIST:

-- select an option --

Submit

Technology ported to CORD-19 by Dimitris Pappas (email: dpappas@aub.gr) and Petros Stavropoulos (email: pstav192@aub.gr). Design & Web Development by Kostas Vrovas (<https://kstavrovas.com/>).
Papers:

Fig. 5.1: In the Covid-19 Search Engine interface the users either write a question or choose from a list of example questions. Then they can filter the results, specifying the section of the articles, which will be retrieved.

⁴<http://cslab241.cs.aueb.gr:5000/>

⁵<https://www.semanticscholar.org/cord19>

⁶<https://www.elastic.co/elasticsearch/>

RESULTS FOR THE QUESTION: PREGNANT WOMEN AND COVID-19

Title: Vertical Transmission of Coronavirus Disease 19 (COVID-19) from Infected Pregnant Mothers to Neonates: A Review

Title: Journal Pre-proof COVID-19 infection among asymptomatic and symptomatic pregnant women: Two weeks of confirmed presentations to an affiliated pair of New York City hospitals COVID-19 infection among asymptomatic and symptomatic pregnant women: Two weeks of confirmed presentations to an affiliated pair of New York City hospitals

Date: 2020-04-09 || Section: Principle Findings

Available on: Doi: 10.1016/j.jogmf.2020.100118

Journal Pre-proof COVID-19 infection among asymptomatic and symptomatic pregnant women: Two weeks of confirmed presentations to an affiliated pair of New York City hospitals COVID-19 infection among asymptomatic and symptomatic pregnant women: Two weeks of confirmed presentations to an affiliated pair of New York City hospitals

We found that COVID-19 infection in pregnant women presenting with obstetric complaints or for delivery is often asymptomatic, suggesting a role for universal testing of pregnant women being admitted to the Labor Unit.

We further found that while many of these women ultimately developed symptoms, disease severity in this small cohort of pregnant patients -86% mild, 9.3% severe, 4.7% critical -appeared similar to what is described in the literature for non-pregnant people.

[7] Results in the context of what is known Our findings are similar to published case series from China of pregnant women with COVID-19 infection that show an overall favorable prognosis.

However, these case series are small [8, 12].

Chen et al [12] described nine cases of pregnant women affected by COVID-19 infection during pregnancy.

None of these patients required ICU admission or mechanical ventilation.

Fig. 5.2: The results of the question are sorted by the article's relevance score and additional info about the article are displayed below the title in gray boxes. The text is highlighted with yellow color, if it contains relevant information or the answer to the question.

5.3.2 BioASQ 8 Challenge

Another contribution that was closely related to our work in this thesis, but is not directly a part of it, is our participation in the BioASQ 8 Challenge. The challenge had two phases, the first was about document and snippet retrieval and the second was about exact answers of factoid, list, yes/no and summary questions.

For the first phase, we used the models from the work that was previously done by Pappas et al. in the BioASQ 7 Challenge (Pappas et al., 2020a), which were further fine-tuned and achieved the best performance amongst its competition. For the second phase, a zero-shot learning approach and trained SpanBERT and BioBERT models were used to predict the exact answers for the factoid and the list instances. The competition was completed before we had implemented the T5 model and therefore we could not evaluate it against the competition.

5.4 Future work

Due to limited resources and time constraints, there were some experiments and aspects of our work that were left as future work. Firstly, the models that were implemented for the BioMRC dataset, were not applied to the Large version of the dataset, due to our limited computational resources.

Furthermore, the experiments with the BERT-based models for solving the BioASQ exact answers task, were not extensive, because both the poor results and the better alterna-

tive options for solving the task (like the T5 model), discouraged us from pushing the configurations to the limit.

A full-context approach of the T5 model, instead of treating each snippet of the exact answer task of the BioASQ challenge as a different instance was tried, but the experiments were not finished due to lack of time. However, the full-context T5 model was implemented and some early testing showed promising results, suggesting that further experimentation and testing could surpass the best results achieved from the current T5 model.

Trying to improve the results of the models on the BioASQ exact answer task, using the BioMRC as a pre-training step, we created a modified BioASQ dataset, where all questions were transformed into cloze-type questions to match the format of the BioMRC dataset. This required an extensive human annotation process, so it was only done for the factoid questions. Further effort could be given, in order to also transform the list questions in the same format and have a complete BioASQ cloze-type dataset for exact answers.

Lastly, we used modified evaluation measures of the BioASQ challenge, to showcase that the current evaluation measures were too harsh and did not account for correct answers by our models. We also compared the results of the modified to the default evaluation measures for a specific T5 model configuration, which indicated a big boost. Yet, we did not manage to run all our experiments using these modified evaluation measures, since that would require more time and computational resources than we had.

Bibliography

- Aronson, Alan R and François-Michel Lang (2010). "An overview of MetaMap: historical perspective and recent advances". In: *Journal of the American Medical Informatics Association* 17.3, pp. 229–236 (cit. on pp. 1, 34).
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun (cit. on pp. 9, 13).
- Beltagy, Iz, Kyle Lo, and Arman Cohan (2019). "SciBERT: Pretrained Language Model for Scientific Text". In: *EMNLP* (cit. on pp. 3, 14, 23, 34).
- Bird, Steven, Loper Edward, and Ewan Klein (2009). *Natural Language Processing with Python*. O'Reilly Media Inc. (cit. on pp. 14, 24).
- Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, et al. (Oct. 2014). "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734 (cit. on pp. 9, 15).
- Craswell, Nick (2009). "Mean Reciprocal Rank". In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, pp. 1703–1703 (cit. on p. 22).
- Cui, Yiming, Zhipeng Chen, Si Wei, et al. (2017). "Attention-over-Attention Neural Networks for Reading Comprehension". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada, pp. 593–602 (cit. on pp. 8, 15, 34).
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1 (Long and Short Papers)* (cit. on pp. 3, 13, 25, 34).

- Dror, Rotem, Gili Baumer, Segev Shlomov, and Roi Reichart (2018). "The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia, pp. 1383–1392 (cit. on pp. 11, 18).
- Hermann, Karl Moritz, Tomáš Košík, Edward Grefenstette, et al. (2015). "Teaching Machines to Read and Comprehend". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. Cambridge, MA, USA, pp. 1693–1701 (cit. on pp. 1, 9, 34).
- Hill, Felix, Antoine Bordes, Sumit Chopra, and Jason Weston (2016). "The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun (cit. on p. 1).
- Huang, Minlie, J. Liu, and Xiaoyan Zhu (2011). "GeneTUKit: a software for document-level gene normalization". In: *Bioinformatics* 27, pp. 1032–1033 (cit. on p. 7).
- Joshi, Mandar, Danqi Chen, Yinhan Liu, et al. (2020). "SpanBERT: Improving Pre-training by Representing and Predicting Spans". In: *Transactions of the Association for Computational Linguistics* 8, pp. 64–77 (cit. on pp. 3, 25, 35).
- Joshi, Mandar, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer (July 2017). "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1601–1611 (cit. on p. 25).
- Kadlec, Rudolf, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst (2016). "Text Understanding with the Attention Sum Reader Network". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany, pp. 908–918 (cit. on pp. 8, 15, 34).
- Kwiatkowski, Tom, Jennimaria Palomaki, Olivia Redfield, et al. (2019). "Natural Questions: A Benchmark for Question Answering Research". In: *Transactions of the Association for Computational Linguistics* 7, pp. 453–466 (cit. on p. 25).
- Leaman, Robert, Rezarta Islamaj Doğan, and Zhiyong Lu (2013). "DNorm: disease name normalization with pairwise learning to rank". In: *Bioinformatics* 29.22, pp. 2909–2917 (cit. on pp. 6, 7).
- Lee, Jinhyuk, Wonjin Yoon, Sungdong Kim, et al. (Sept. 2019). "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4, pp. 1234–1240 (cit. on pp. 3, 6, 25, 35).
- Li, J., Yueping Sun, Robin J. Johnson, et al. (2016). "BioCreative V CDR task corpus: a resource for chemical disease relation extraction". In: *Database: The Journal of Biological Databases and Curation* 2016 (cit. on p. 14).

- Lindberg, Donald A. B., Betsy L. Humphreys, and Alexa T. McCray (1993). "The Unified Medical Language System." In: *Yearbook of medical informatics* 1, pp. 41–51 (cit. on p. 1).
- Lipscomb, C. E. (2000). "Medical Subject Headings (MeSH)." In: *Bulletin of the Medical Library Association* 88 3, pp. 265–6 (cit. on p. 7).
- Nye, Benjamin E., Junyi Jessy Li, Roma Patel, et al. (2018). "A Corpus with Multi-Level Annotations of Patients, Interventions and Outcomes to Support Language Processing for Medical Literature". In: *Proceedings of the conference. Association for Computational Linguistics. Meeting 2018*, pp. 197–207 (cit. on p. 14).
- Pappas, Dimitris, Ion Androutsopoulos, and Haris Papageorgiou (2018). "BioRead: A New Dataset for Biomedical Reading Comprehension". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan (cit. on pp. 1, 6, 19, 34).
- Pappas, Dimitris, Ryan McDonald, Georgios-Ioannis Brokos, and Ion Androutsopoulos (2020a). "AUEB at BioASQ 7: Document and Snippet Retrieval". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Peggy Cellier and Kurt Driessens. Cham: Springer International Publishing, pp. 607–623 (cit. on pp. 37, 38).
- Pappas, Dimitris, Petros Stavropoulos, Ion Androutsopoulos, and Ryan McDonald (July 2020b). "BioMRC: A Dataset for Biomedical Machine Reading Comprehension". In: *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*. Online: Association for Computational Linguistics, pp. 140–149 (cit. on pp. 2, 6, 13, 17, 22, 34).
- Ra'el, Colin, Noam Shazeer, Adam Roberts, et al. (2020). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140, pp. 1–67 (cit. on pp. 3, 27, 35).
- Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang (2016). "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas, pp. 2383–2392 (cit. on pp. 1, 2, 25, 35).
- Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016). "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany, pp. 1715–1725 (cit. on p. 13).
- Soysal, Ergin, Jingqi Wang, Min Jiang, et al. (2017). "CLAMP – a toolkit for efficiently building customized clinical natural language processing pipelines". In: *Journal of the American Medical Informatics Association* 25.3, pp. 331–336 (cit. on p. 1).
- Šuster, Simon and Walter Daelemans (2018). "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 (NAACL-HLT), Volume 1 (Long Papers)*. New Orleans, Louisiana, pp. 1551–1563 (cit. on p. 1).

- Taboureau, O., Sonny Kim Nielsen, K. Audouze, et al. (2011). "ChemProt: a disease chemical biology database". In: *Nucleic Acids Research* 39, pp. D367–D372 (cit. on p. 14).
- Taylor, Wilson L. (1953). "'Cloze Procedure': A New Tool for Measuring Readability". In: *Journalism Quarterly* 30.4, pp. 415–433 (cit. on pp. 1, 9, 23).
- Trischler, Adam, Tong Wang, Xingdi Yuan, et al. (2017). "NewsQA: A Machine Comprehension Dataset". In: *Proceedings of the 2nd Workshop on Representation Learning for NLP*. Vancouver, Canada, pp. 191–200 (cit. on p. 25).
- Tsatsaronis, G., G. Balikas, P. Malakasiotis, et al. (2015). "An Overview of the BioASQ Large-Scale Biomedical Semantic Indexing and Question Answering Competition". In: *BMC Bioinformatics* 16.138 (cit. on pp. 2, 22, 34).
- Vandesompele, J., K. De Preter, Filip Pattyn, et al. (2001). "Accurate normalization of real-time quantitative RT-PCR data by geometric averaging of multiple internal control genes". In: *Genome Biology* 3, research0034.1–research0034.11 (cit. on p. 7).
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, et al. Curran Associates, Inc., pp. 5998–6008 (cit. on pp. 13, 27, 36).
- Wei, Chih-Hsuan, B. Harris, Hung-Yu Kao, and Zhiyong Lu (2013). "tmVar: a text mining approach for extracting sequence variants in biomedical literature". In: *Bioinformatics* 29 11, pp. 1433–9 (cit. on p. 7).
- Wei, Chih-Hsuan, Bethany R. Harris, Donghui Li, et al. (2012a). "Accelerating literature curation with text-mining tools: a case study of using PubTator to curate genes in PubMed abstracts". In: *Database* 2012 (cit. on pp. 7, 24, 34).
- Wei, Chih-Hsuan, Hung-Yu Kao, and Zhiyong Lu (2012b). "SR4GN: A Species Recognition Software Tool for Gene Normalization". In: *PLoS ONE* 7 (cit. on p. 7).
- Wu, Y., Mike Schuster, Z. Chen, et al. (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: *ArXiv abs/1609.08144* (cit. on pp. 13, 14).
- Zhu, Y., Ryan Kiros, R. Zemel, et al. (2015). "Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books". In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 19–27 (cit. on p. 14).

List of Figures

2.1	Overview of the ASReader model.	10
2.2	Overview of the AOARReader model.	11
2.3	Example passage-question instance of BioMRC. The passage is the abstract of an article, with biomedical entities replaced by @entityN pseudo-identifiers. The original entity names are shown in square brackets. Both 'edematous' and 'edema' are replaced by '@entity4', because Pubtator considers them synonyms. The question is the title of the article, with a biomedical entity replaced by . @entity0 is the correct answer.	11
2.4	Example from BioMRC Tiny. In Setting A, humans see both the pseudo-identifiers (@entityN) and the original names of the biomedical entities (shown in square brackets). Models see only the pseudo-identifiers, but the pseudo-identifiers have global scope over all instances, which allows the models, at least in principle, to learn entity properties from the entire training set. In Setting B, humans no longer see the original names of the entities, and models see only the pseudo-identifiers with local scope (numbering reset per passage-question instance).	12
3.1	Illustration of our SciBERT-based models. Each sentence of the passage is concatenated with the question and fed to SciBERT. The top-level embedding produced by SciBERT for the first sub-token of each candidate answer is concatenated with the top-level embedding of [MASK] (which replaces the placeholder) of the question, and they are fed to an MLP, which produces the score of the candidate answer. In SciBERT-Sum-Reader, the scores of multiple occurrences of the same candidate are summed, whereas SciBERT-Max-Reader takes their maximum.	17
3.2	More detailed statistics and results on the development subset of BioMRC Lite. Number of passage-question instances with 2, 3, ..., 20 candidate answers (top left). Accuracy (%) of the basic baselines (top right). Accuracy (%) of the neural models in Settings A (bottom left) and B (bottom right).	19
3.3	HTML sample from the Setting A BioMRC tiny dataset for the human evaluation. The human evaluator picks one entity and it is highlighted across the document (with red color). Then the evaluator chooses whether he/she could answer the question using prior knowledge, the context, or both and submits his/her answer.	20

5.1	In the Covid-19 Search Engine interface the users either write a question or choose from a list of example questions. Then they can filter the results, specifying the section of the articles, which will be retrieved.	37
5.2	The results of the question are sorted by the article's relevance score and additional info about the article are displayed below the title in gray boxes. The text is highlighted with yellow color, if it contains relevant information or the answer to the question.	38

List of Tables

2.1	Statistics of BioMRC Large. All lengths are measured in tokens using a whitespace tokenizer.	8
2.2	Statistics of BioMRC Lite. All lengths are measured in tokens using a whitespace tokenizer.	9
2.3	Statistics of BioMRC Tiny. The questions were answered by humans. All lengths are measured in tokens using a whitespace tokenizer.	9
3.1	Training, development, test accuracy (%) on BioMRC Lite in Setting A (global scope of entity identifiers), training times (epochs \times time per epoch), and number of trainable parameters (total, word embedding parameters, entity identifier embedding parameters).	18
3.2	Training, development, test accuracy (%) on BioMRC Lite in Setting B (local scope), training times (epochs \times time per epoch), and number of trainable parameters (total, word embedding parameters, entity identifier embedding parameters).	18
3.3	Human agreement (Cohen's Kappa, %) on BioMRC tiny. Avg. pairwise scores for non-experts.	20
3.4	Accuracy (%) on BioMRC tiny. Best human and model scores shown in bold.	21
4.1	Results of the T5 model using the post-processing function and without using the custom scoring function.	30
4.2	Results of the T5 model without using the post-processing function and without using the custom scoring function.	30
4.3	Results of the T5 model without using the post-processing function and with our custom scoring function.	31
4.4	Problems of various question-answer instances from the BioASQ dataset. The predictions of the T5 model are also showed in the table. These predictions were incorrect using the BioASQ evaluation measures, but correct using our custom evaluation measures. The last column mentions the problem with each instance.	32
4.5	Results of T5 trained on BioASQ dataset using the original BioASQ evaluation measures and the custom evaluation measures using the fuzzywuzzy module.	33