# Machine Comprehension for Clinical Case Reports

by

Mai Phuong Pham

S.B. in Computer Science and Molecular Biology, Massachusetts Institute of Technology (2019)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Molecular Biology

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020



Author .....
Department of Electrical Engineering and Computer Science
May 12, 2020

Certified by .....
Peter Szolovits
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

**Machine Comprehension for Clinical Case Reports**

by

Mai Phuong Pham

Submitted to the Department of Electrical Engineering and Computer Science
on May 12, 2020, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Molecular Biology

## Abstract

Over the past decade, question answering (QA) has been an active area of research in natural language processing (NLP). Despite much progress in general knowledge tasks, question answering in specialized domains, such as healthcare and medicine, hasn't seen a breakthrough due to the lack of large, reliable datasets. Moreover, using Mechanican Turk participants to ask questions about the texts, a common approach taken by general knowledge QA datasets, is often not applicable to specialized domains due to the complexity of the texts and the need for specialized knowledge. Introduced in 2018, Clinical Case Report (CliCR, [24]) is one of a few QA datasets in the medical domain. The dataset used BMJ clinical case reports with more than 100,000 gap-filling queries about these cases. The lack of human-formed natural questions is a challenge for this dataset, as well as the generalizability of trained NLP models on it.

This thesis attempts different approaches to the question answering task on gap-filling queries. Besides frameworks designed specifically for filling-in-the-blanks tasks, I show that systematic modifications on the queries will allow other approaches, such as language models, to outperform conventional approaches. The BioBert QA model ([14]) achieves 55.2 exact match (EM) accuracy and 59.8 F1 score on CliCR, higher than the current best performer, gated-attention machine reader (EM=22.2, F1=32.2, [6]) and human expert readers (EM=35, F1=53.7, [24]).

Moreover, this work seeks to understand if language models, such as BioBert ([14]), focus on basic linguistic elements of a question (Wh- question words, cloze position, and question mark). Through a series of experiments across 3 different QA datasets and visualization of trained attention heads, some weak attention patterns are identified. However, when combined with further analysis on the role of question words in QA task, it becomes clear that BERT models might not focus on question words or cloze position, and question mark. Future extension of this thesis should seek to understand the role of questions in the QA task using language models.

Thesis Supervisor: Peter Szolovits
Title: Professor of Electrical Engineering and Computer Science

# Acknowledgments

Completing this thesis amid a global pandemic is one of the major achievements in my life. The research journey has been both intellectually and emotionally challenging. I have learned that I am capable of learning natural language processing on my own in the last nine months. However, there were a lot of moments when I had doubted my ability to conduct independent research. Although this master thesis is not necessarily a breakthrough, I am glad I have completed it. With this opportunity, I would like to thank many people in my life, without whom I could not have achieved and finished my thesis.

My advisor, Peter Szolovits, has always been thoughtful and incredibly supportive of this journey. Pete has inspired me to become an independent researcher and encouraged me to set a high standard for my thesis. I realized that, sometimes, a lot more could be achieved with a little push.

It has been a great experience to immerse myself in the Clinical Decision-Making group. I am fortunate to get to know other graduate students, especially Geeticka, Tiffany, Willie, and Matthew. Thank you for your time and support throughout these past few months.

Family and friends have been a constant source of support. I am thankful for my partner, Jonathan Oh, for always being by my side and sharing my stress and concern. My parents have become a great source of encouragement. Other MIT friends, Tue Vo, Maggie Wu, Nancy Wang, Diana Molodan, and Diane Zhou have made my last semester as a student a lot more enjoyable.

Although there are many uncertainties in the near future, I feel fortunate to have been part of the MIT community, and now a soon-to-be MIT alumna. The past five years at MIT have shaped me into who I am today. No matter where I will be after MIT, I am inspired to bring my knowledge and skills I have learned here to contribute to a great good in the world.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

*This chapter motivates a better question answering (QA) model for the Clinical Case Reports (CliCR, [24]) dataset. It also provides an overview of current approaches in creating QA datasets and modeling of QA tasks to show the breadth of the field in the past decade.*

## 1.1 Motivation

Over the past decade, question answering (QA) has been an active area of research in natural language processing (NLP). Despite much progress in open-domain QA, QA tasks in specialized domains like healthcare and medicine receive little attention so far due to the lack of reliable, large-scale datasets. Nonetheless, this domain has tremendous potential for applications of QA. As an example, machine comprehension can be used in the clinical workflow to help doctors research external medical knowledge or answer patient-specific questions. Combined with knowledge graph document retrieval systems [3], machine reading at scale has the potential to be incorporated into the clinical workflow in the future.

Introduced in 2018, Clinical Case Report (CliCR, [24]) is one of a few QA datasets in the medical domain. The dataset used BMJ clinical case reports with more than 100,000 gap-filling queries about around 12,000 cases. The length and complexity of the documents, as well as the lack of human-formed natural questions are the

challenges for this dataset, making it more difficult to generalize pre-trained NLP models. This thesis aims to explore different approaches to QA task on CliCR. These approaches include both conventional method designed for fill-in-the-blank task, as well as modifying language models for QA framework. It concludes with an analysis to show how language model might or might not utilize basic elements of a query, such as cloze position, question words, or question mark '?' tokens.

This chapter will be followed by an introduction to the question answering task in natural language processing (section 1.2) and a comprehensive literature review which details past datasets and approaches to QA tasks (section 1.3). The chapter concludes by providing the contributions of this thesis to improve machine comprehension on the CliCR dataset, and the analysis of how language models might not interpret questions using basic linguistic elements, such as question words and question mark, the same way human readers do.

## 1.2 Question Answering Task

Question answering (QA) is a task in which the system reads text passages and then answers relevant questions. QA tasks heavily depend on the format of the datasets, which can be generalized into three main types:

- Natural question answering: machine selects text span from supporting document or selects among candidates to **answer a natural question** relevant to the document. Datasets of this type include SQuAD [19, 18], MCTest [20], and WikiQA [28].

- Gap filling: machine selects answer span from supporting document or selects the best answer among candidate answers to **fill in the blank in a query** based on its supporting document. Datasets of this type are often called **cloze dataset**, and include CNN/Daily Mail [8] and CliCR [24].

- Conversation: machine answers user's question based on many different sources of information.

Table 1.1 provides examples for each of these tasks.

## 1.3 Literature Review

### 1.3.1 Question Answering

Over the past decade, there has been much progress in all three different QA tasks, fueled by large datasets, models, and leaderboards [22]. Recently, the natural language processing (NLP) community has shifted to using pre-trained models and embeddings to exploit the benefits of transfer learning [5, 29, 14, 1, 13, 15, 11, 17]. These pre-trained models have achieved state-of-the-art in many QA datasets, most of which are open-domain and built from news, stories, and Wikipedia texts.

Question answering datasets are collections of (passage, query, answer) triples. Text passages contain the answer to the query, in exact words or non-exact words. Query can be an interrogative sentence (e.g. *"What is the capital city of France"*) or a fill-in-the-blank sentence (e.g. *"The capital city of France is _."*). The answer can be a single token or multiple tokens, either chosen from a set of candidate answers or generated from the support passages. The CliCR dataset falls into the **cloze dataset** category. This type of dataset requires readers to choose from a number of candidate answers to fill in a blank in the query. Oftentimes, the query is a summarization or paraphrase of the supporting document. Therefore, this type of dataset allows us to test for a different reading comprehension ability, as compared to interrogative question-answer. Similar datasets in this category include CNN/Daily Mail [8], Children's Book Test (CBT) [9], and RACE [12].

Machine comprehension on cloze datasets has been an active research area over the past few years. The leading CNN/Daily Mail leaderboard model is Gated Attention and Memory as Acyclic Graph Encoding (GA+MAGE) [7]. The Gated-Attention Reader (GA) [6] performs multiple hops over the document while maintaining the multiplicative interactions between query and the contextual embeddings. This mechanism mimics the way human readers keep the question in mind during multiple passes

| Task | Example datasets | Example query |
|---|---|---|
| Natural question answering | SQuAD | **Context:** [...]The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. [...] **Question:** Which NFL team represented the AFC at Super Bowl 50? **Answer:** *Denver Broncos* |
| Gap filling | CNN/Dailymail | **Context:** [...] Japan officially agreed in February to lend up to 100 billion dollars to the IMF to provide financial lifelines to emerging economies hit hard by the worldwide downturn. US Treasury Secretary Timothy Geithner has said President Barack Obama would discuss new global financial regulatory standards at the London summit. [...] **Question:** US President Barack Obama will push higher financial regulatory standards for across the globe at the upcoming G20 summit in London, *@placeholder* said Thursday. **Answer:** *Timothy Geithner* |
| Conversation | | **Context:** A wide collections of task-specific training datasets, and knowledge graphs. **Question:** Okay Google, how is the weather today? **Answer:** Today in Cambridge expects the high of 9 and low of 3. |

Table 1.1: Three main types of QA tasks: Natural question answering, gap filling, conversation

of reading to filter out irrelevant information to the query.

Since the introduction of the transformer in 2017 [25], many deep learning models based on this architecture have gained much traction and achieved state-of-the-art accuracy on many machine comprehension datasets. For example, OpenAI GPT-2 [17] achieved a 93.3% accuracy on the CBT dataset, closely matching human's performance (about 95% accuracy). Pre-trained models, such as GPT-2 [17], BERT [5], and XLNet [29], have demonstrated the ability to learn sequences of natural language from a vast amount of training data and performed very well on many zero-shot learning tasks. Not only do these models perform well on open-domain tasks, but their variants, such as BioBert [14] or ClinicalBert [1], have also demonstrated state-of-the-art performance on healthcare or medicine related NLP tasks, which further elaborates the potentials of transfer learning.

**For the CliCR dataset,** at the time of the proposal, the best model is Gated-Attention Reader [24] with *word2vec* embeddings trained on PubMed abstracts with over 9 billions tokens. This model is examined with three different setups for candidate answers: marked entities, anonymized entities, no entities marked (in which the reader will search through the entire passage for the answer and explicitly indicate the answer positions in the passage) and attains the highest F1 score of 33.9 for no entities marked. On the human reader benchmark, a novice reader (having some linguistic knowledge, but no medical knowledge) has an EM of 31 and an F1 of 45.1. An expert reader (having both linguistic and medical knowledge) has both higher EM ($EM = 35$) and F1 ($F1 = 53.7$). The human readers spent 15 minutes on average to read the passage and answer the query (about 2 to 3 minutes per query). Although this result is much lower than the theoretical bound of F1 for a human reader (F1 bound of 100), this can be attributed to the complexity of the reading, answer openness (many possible correct answers and some answers are not verbatim in the passage), and the unnatural format of the gap-filling task.

### 1.3.2 Study of Language Model's Attention Mechanism

The attention mechanism of language models has gained much traction recently. Many studies [4, 26] have attempted to study the roles of attention heads in Transformers and made connections to known linguistic structures. Clark et al. [4] study of BERT on a dataset of 1000 random Wikipedia consecutive paragraphs (max sequence length 128) found that while some attention heads have signature attention mechanisms (attention to next token/previous token, attention to `[CLS]`, attention to part of speech such as verbs and pronouns), many other attention heads seem to attribute attention more randomly, as measured through entropy of the attention distribution. This observation is supported by Vig et al. study of GPT-2 [26]. Vig et al. found that deeper layers have more specific attention patterns, and often pay attention to particular dependency types, such as subjects, auxiliaries, and conjunctions.

Clark et al. argues that many attention heads pay attention to `[CLS]` to propagate context information to the new representation. Although a similar observation was made for `[SEP]`, they argue that `[SEP]` might have acted as a *placeholder* for attention (ie. no other good places to pay attention to). Vig et al. argues that both `[CLS]` and `[SEP]` can act as *placeholders*. The roles of special tokens in language models remain controversial.

## 1.4 Goals and contribution

Given the lack of studies of question answering tasks in the medical domain and their potential to be incorporated in the medical workflow, this thesis aims to tackle a QA task on CliCR, a large scale QA dataset about clinical case reports. The contribution of this work can be summarized as follows:

1. Achieve state-of-the-art QA performance on CliCR using a number of different approaches (including a gap-filling framework and language models).

2. Show that a systematic modification to a cloze dataset allows language models

for the QA task to achieve high performance on gap-filling task.

3. Show how language models might not interpret questions based on basic linguistic elements, such as question words and question marks, the way human readers do.

# Chapter 2

# Datasets

*This chapter introduces the main data set of this thesis, CliCR, and two standard datasets of the QA domain, SQuAD v1.1 and SQuAD v2.0. Since the aim of this thesis is to identify state-of-the-art NLP model for the CliCR dataset, this dataset will be evaluated for the QA task. SQuAD v1.1 and SQuAD v2.0 will be used to evaluate the role of Wh- words in questions. CliCR will be studied to evaluate the attention weights of BERT and ALBERT models in basic elements of cloze questions. Details about these datasets will be valuable to interpret the results in Chapter 4.*

## 2.1 Clinical Case Report (CliCR)

The Clinical Case Report (CliCR) dataset was introduced in 2018 by Simon Suster and Walter Daelemans [24]. The dataset was created from 11,846 BMJ Case Reports, spanning the years from 2005 to 2016. A case report is a detailed description of a clinical case focusing on rare diseases, unusual presentation of common conditions and novel treatment methods (Table 2.2). Each case report has a *Learning point*, a summary paragraph that paraphrases the key pieces of information from that report. The queries are sentences from the *Learning point* with medical entities blanked out. One example of such query is shown in Table 2.1.

During data processing, the authors removed HTML boilerplate from the crawled

| Dataset | Example |
|---|---|
| CliCR | **Passage**<br>[...] A gradual improvement clinical and laboratory status was achieved within 20 days of antituberculous treatment. The patient was then subjected to a thoracic CT scan that also showed significant radiological improvement. *Thereafter, tapering of corticosteroids was initiated with no clinical relapse.* The patient was discharged after being treated for a total of 30 days and continued receiving antituberculous therapy with no reported problems for a total of 6 months under the supervision of his hometown physicians. [...]<br>**Query**<br>If steroids are used, great caution should be exercised on their gradual tapering to avoid *@placeholder*.<br>**Answer**<br>relapse |
| SQuADv1.1 | **Passage**<br>[...] *In meterology, precipitation in any product of the condensation of atmospheric water vapor that falls under gravity.* The main forms of precipitation include drizzle, rain, sleet, snow, graupel, and hail. [...]<br>**Query**<br>What causes precipitation to fall?<br>**Answer**<br>gravity |
| SQuADv2.0 | **Passage**<br>[...] Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. *These later laws had a low cost to society - the species were relatively rare - and little opposition was raised.*<br>**Question**<br>What laws faced significant opposition?<br>**Plausible answer**<br>later laws |

Table 2.1: Question-answer pairs in CliCR, SQuADv1.1 and SQuADv2.0. Answer texts are colored to match with the segments in the passage. The surrounding contexts are *italic*.

| Topics | % | Example |
|---|---|---|
| Problem | 67 | tuberculosis, abdominal pain, acute myocardial infarction |
| Treatment | 22 | chemotherapy, surgical intervention, vitamin D supplement |
| Test | 11 | MRI, histopathological exam |

Table 2.2: Topics of case reports statistics in CliCR dataset

reports using jusText [16], segmented and tokenized the texts with cTakes [21], and annotated the medical entities using Clamp [23]. In order to minimize error in the entity recognition process, the authors conducted manual tests, and selected a concept unique identifier (CUI) for a valid entity, which links it to the UMLS Metathesaurus [10]. The queries that were blanked out incorrectly (wrongly recognized entities found during manual test or non-existent in UMLS Metathesaurus) were removed. The final dataset contains 104,919 gap-filling queries for about 11,846 passages (around 1800 tokens on average), with single-token or multiple-token answers. Figure B-1 shows the distribution of length of passages, queries, and answers in the CliCR dataset. The length of passages has a bimodal distribution because the passages were shorter before 2008 [24]. Overall, as compared to the SQuAD datasets (Figure B-2), CliCR has longer support passages, queries, and answers.

The training, development, and testing sets are created by splitting these 11,846 passages and their corresponding queries randomly into 3 subsets following a 90:5:5 ratio, as in [24]. Splitting on the passage-level, instead of query-level, ensures that the model cannot "cheat" by returning details in the training set during evaluation. Similar to SQuAD v2.0 (Section 2.3), some queries in CliCR don't have answers verbatim in the support passages. The answers to these queries are relevant and can be inferred from the passage. Table 2.3 details basic data statistics about the train, development, and test sets, including the number of passages and queries, as well as the percentage of queries with answers verbatim in the passages.

| Dataset | Number of cases | Number of queries | % of answers verbatim in passage |
|---|---|---|---|
| Train | 10,638 | 91,344 | 61.6 |
| Dev | 584 | 6,391 | 60.8 |
| Test | 624 | 7,184 | 59.8 |

Table 2.3: Data statistics for train, development, and test sets (CliCR)

## 2.2 Stanford Question Answering Dataset (SQuAD) v1.1

Introduced in 2016, SQuAD v1.1 [19] is an open-domain QA dataset, which requires the reader to select an answer from a large collection of documents over a wide range of topics. The SQuAD v1.1 dataset was created from 536 Wikipedia articles (23,215 paragraphs) with questions posed by Amazon Mechincal Turkers (107,785 questions). For each paragraph, the turkers had to ask and answer up to 5 questions. The question was entered in a text field and answer was highlighted in the paragraph. The next round, a different group of turkers, given the initial dataset and the questions, were asked to select a span of text in the paragraph that contained the answers. In this first version of the SQuAD dataset, all questions must have at least one answer. Examples of question-answer pairs are shown in Table 2.1.

Processed articles were partitioned randomly into training set (80%), development set (10%), and test set (10%). Figure B-2 shows the distribution of length (in tokens) of passages, queries, and answers in the SQuAD dataset. On average, a SQuAD passage is 152 tokens, question is 12 tokens, and answer is 4 tokens long, much shorter than CliCR.

## 2.3 Stanford Question Answering Dataset (SQuAD) v2.0

SQuAD v2.0 [18] augments 53,775 new harder questions without explicit answers for the same documents in SQuAD v1.1 (about 40% of all queries). These questions are posed so that (1) they are relevant to the paragraph, and (2) the paragraph contains

plausible answers for them. These answers often require more inference capability, thus making SQuAD v2.0 more challenging. An example of such questions is shown in Table 2.1. SQuAD v2.0 has about 60% of queries with answers appearing verbatim in the passage, making it similar to CliCR in the distribution of answerable versus unanswerable questions.

One of the most straightforward ways to build a QA system is to train it to identify a specific span in the available text that answers a question. If the expected answer is verbatim in the document, the success of such QA model can be judged by the exact match criterion (the model identifies the exact text span that is the answer). This approach is commonly adopted for QA task in SQuADv1.1 and SQuADv2.0. However, it cannot handle two common cases in real application of question answering where the answers are not verbatim in the supporting documents. First is the possibility that the answer is not mentioned in the supporting document (for example, in unanswerable queries in SQuADv2.0). In addition, because of a sliding window selection of text imposed by certain methods, the window may fail to contain the answer although the overall text does. Second, in real question answering, the answers only need to be semantically correct, not necessarily an exact token match. For example, if the text says “interval CT scans of the chest” and the cloze answer is "chest CT", we should consider “chest CT” as just as valid an answer as “interval CT scans of the chest” (even though the exact phrase "chest CT" does not appear in the text). The approach taken by the influential SQuAD2.0 corpus was to identify only textually identical words/phrases as correct, which excludes both these alternative. Indeed, in that approach, NULL would be considered the correct answer to both.

# Chapter 3

# Methodology

*This chapter provides the methodology for the two main goals of this thesis, question answering (QA) and study of attention head on CliCR. In QA, three main model frameworks, attention sum model (AS), gated attention model (GA), and pretrained language models (LMs) for QA, are explored. LMs are complex models that have been self-supervised pretrained on a large number of documents, and thus, are able to capture a wide variety of natural language information (Section 3.2.3). LMs are used directly in AS to generate context-dependent query and document embeddings. AS uses a score function of cloze position and query embeddings to select the best answers among candidates (Section 3.2.1). The second model framework, GA, mimics the way a human reader finds query-relevant information on the passage. It aims to learn query-informed, contextualized document embeddings. These embeddings are used to select the best answer for the cloze position among candidate answers (Section 3.2.2). In the LMs-for-QA model, LMs are fine-tuned to generate answers by selecting a text span within the passage (Section 3.2.3). Section 3.1 contains details about data processing for various tasks. Section 3.2 details model architectures and evaluation for the QA task. The last section of this chapter details the study of attention heads to help understand some attention patterns exhibited by LMs.*

## 3.1 Data Processing

In the original CliCR dataset, each example consists of a title, a passage, a passage ID, and multiple question-answer-ID triples. Entities in the passages and questions are tagged with entity tags `BEG__ __END`. There are 3 `json` files for train, development, and test sets, following the same splitting as seen in the paper [24].

**For models relying on pretrained LMs**, each data point has the general form of `<title, context, question, answer, position of answer, isimpossible>` where `context` is a small spanning text within the original passage. In SQuAD datasets, most of the time, `context` is the paragraph from which turkers posed questions. Because there is a limit in the maximum sequence length in pretrained LMs, for longer paragraphs in SQuADs and all of the passages in CliCR, data points are formed with smaller `contexts` by dividing the original passage into overlapping texts using a **sliding window** technique (implemented in the HuggingFace's Transformer library in SquadProcessor [27]). As a result, even if the answer is verbatim in the original passage, it is not guaranteed to be in some `contexts`. When that is the case (i.e `context` doesn't contain the answer), `isimpossible` is `True`, and `False` otherwise.

**For LMs-for-QA model** (Section 3.2.3), the blank is replaced with `[MASK]` or some *Wh- question words*, such as *What, Which* and *Who*. **For AS model** (Section 3.2.1), the blank in the question is replaced with $[MASK]_1[MASK]_2$ where $[MASK]_1$ denotes the start and $[MASK]_2$ denotes the end of the answer. Each data point of AS also has a list of `candidate answers`, similar to GA.

For the **Study of Attention Head** (Section 3.3), each data point has the form `<context, question>`. The `context` is generated from the original passage by the sliding window technique discussed above, and *@placeholder* in the `question` is filled with `[MASK]`.

## 3.2 Question Answering Tasks on CliCR

### 3.2.1 Attention Sum Model (Baseline)

Using the `biobert_v1.1_pubmed` pretrained model, contextualized embeddings are generated for the query and the context. A data point will take the form `<[CLS] query [SEP] context [SEP]>`. As discussed in Section 3.1, a sliding window approach divides the passage into many context-overlapping data points. The embeddings of the $[\texttt{MASK}]_1[\texttt{MASK}]_2$ are averaged across those data points. Embeddings of tokens in the overlapping part of two contexts are chosen to maximize context (more details in Section 3.2.3).

For a candidate answer that spans from token `a` to token `b`, let $T_a$ and $T_b$ be their embeddings, respectively, then the probability of that text span being the answer is computed as $Pr(a...b) = T_{[MASK]_1} \cdot T_a + T_{[MASK]_2} \cdot T_b$. The probability of a candidate answer being the correct answer is proportional to the sum of the probability of the candidate's occurences. The candidate with the highest probability will be chosen as the predicted answer.

This model doesn't require any training on CliCR as the embeddings are generated from a pretrained language model. For its simplicity, AS is chosen to be the baseline model.

### 3.2.2 Gated Attention Model

*This section aims to give a short explanation of the GA model. More details can be found in the original paper "Gated-Attention Readers for Text Comprehension" [6].*

The Gated Attention Model (GA) [6] consists of multiple Bi-directional Gated Recurrent Units (Bi-GRUs) [2]. Its model architecture is illustrated in Figure 3-1. The model aims to generate contextualized embeddings for the document and the query over K layers. At each layer, GA updates the document embedding based on the query. The document embedding and query embedding are transformed by taking the output of corresponding document and query Bi-GRUs. Then, the input for the

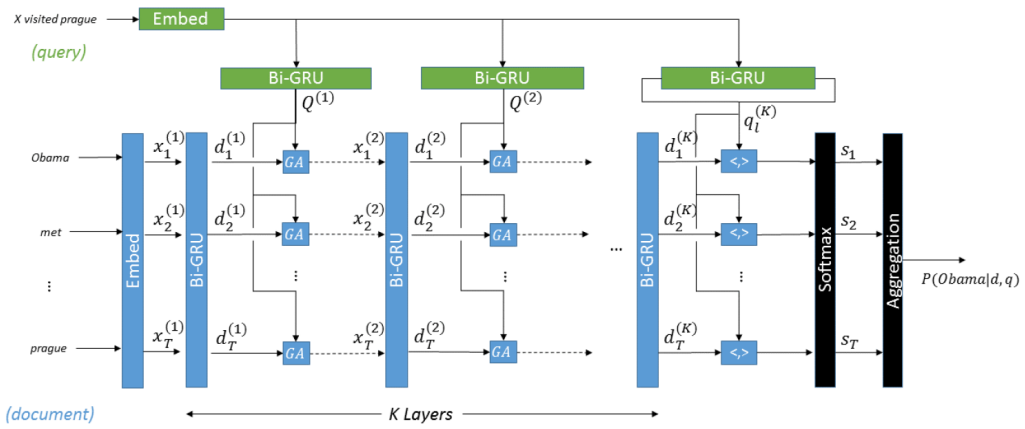



Figure 3-1: GA Model Architecture. *Figure adopted from Dhingra et al.* [6]

next layer is generated as $X^{(k)} = QA(D^{(k)}, Q^{(k)})$ through a $QA(.)$ module. For each token of the document, this $QA(.)$ module forms a token-specific representation of the query, using soft attention, and then multiplies the query representation element-wise with the document token representation. This process mimics how a human reader typically reads the query, then narrows the search for the answer by paying attention to relevant information in the document.

In the final layer, let $q_l^{(K)}$ be the final output of the query Bi-GRU at position $l$, the position of `@placeholder` (blank or cloze position), and $D^{(K)}$ be the final output of the document Bi-GRU. The probability that a token in the document answers the query is computed as:

$$s = softmax(q_l^{(K)} \cdot D^{(K)})$$

The probability of a candidate answer $c \in \mathcal{C}$ being the correct answer is aggregated over all document's tokens that appears in $c$ and renormalized over the candidates

$$Pr(c|d, q) \propto \sum_{i \in \mathcal{S}(c,d)} s_i$$

where $\mathcal{S}(c, d)$ is the set of positions where a token in document $d$ appears in $c$. A candidate with the maximum probability is chosen

$$a^* = argmax_{c \in \mathcal{C}} Pr(c|d, q)$$

In this work, I modified the GA model so that it can handle the case when the true answer does not appear verbatim in the document. The first entity in the document (`@entity0`) is reserved for such cases. The full implementation in Pytorch of this GA model can be found on my Github repository [1].

### 3.2.3 Pretrained Language Models

*This section reviews two language models, BERT [5] and ALBERT [13], as the reviews will be critical to the Study of Attention Head (Section 3.5). This work also explores other language models, BioBERT [14], XLNet [29]. Their architecture details can be found in the corresponding papers.*

#### Bidirectional Encoder Representations from Transformers (BERT)

BERT [5] is a pretrained language model introduced in 2019 by Devlin et al. Its model architecture consists of multi-layer bidirectional Transformer encoders, as described in Vaswani et al. [25]. There are two main steps in the BERT framework: pretraining and fine-tuning. During pre-training, the model is trained on unlabeled data for two pretraining tasks, masked language model (LM) and next sentence prediction (NSP). For fine-tuning, the BERT model is initialized with the pretrained parameters, and the parameters will be fine-tuned with labeled data from the downstream tasks. Figure 3-2 presents a high-level overview of the BERT framework during pretraining and fine-tuning.

To accomplish a wide variety of downstream tasks, the **BERT input representation** is able to unambiguously represent both a single sentence and a pair of sentences. In many cases, a "sentence" is actually a span of continuous text rather than a linguistic sentence, for example in `<Question, Answer>`. Every sequence starts with a special token `[CLS]`, whose final hidden state is treated as the aggregate sequence representation. A pair of sequences is separated with a special token `[SEP]`. Overall, the input representation consists of token embeddings, segment embeddings,

[1] Code is available at https://github.com/phuongpm241/thesis_work/tree/master/ga_model.

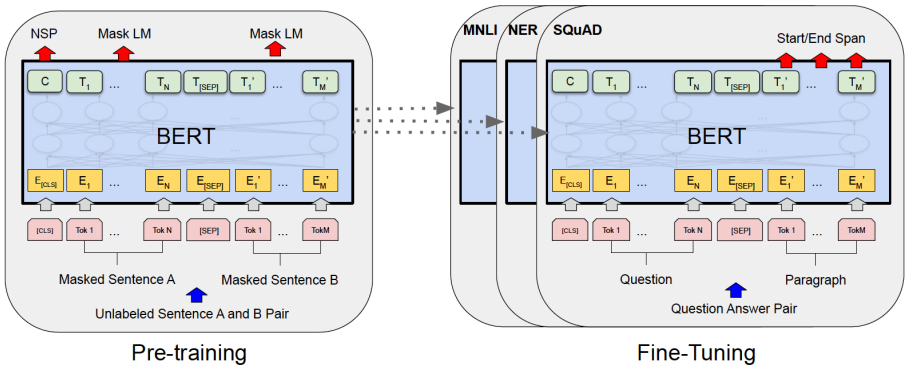



Figure 3-2: High-level pretraining and fine-tuning framework for BERT. Except for the output layers, the same architecture is used in both pretraining and finetuning. During fine-tuning, the parameters are initialized with their pretrained values, and all parameters are fine-tuned. *Figure adopted from Devlin et al.* [5].

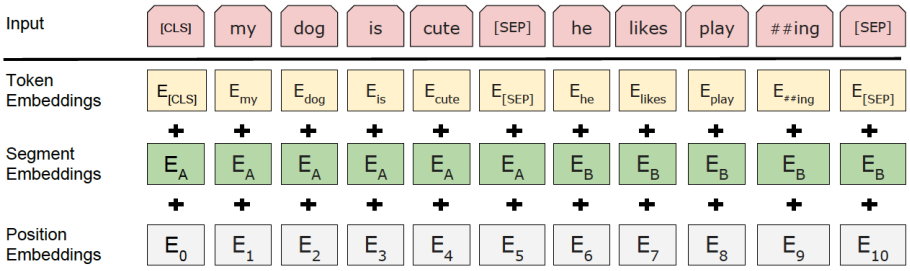



Figure 3-3: An example of BERT input representation. The input embeddings consists of token embeddings, segment embeddings, and position embeddings. *Figure adopted from Devlin et al.* [5].

and position embeddings. Figure 3-3 gives an example of such a representation.

**The pretrained tasks** include Masked LM and NSP. During Masked LM, 15% of the token positions are chosen at random for prediction. To prevent discrepancy between pretraining and fine-tuning datasets, as the fine-tuning dataset is unlikely to contain the `[MASK]` token, the target token is only replaced with (1) `[MASK]` 80% of the time, (2) a random token 10% of the time, and (3) itself the remaining 10% of the time.

During NSP, the training set is composed of sentence pairs `<A, B>` where 50% of the time B is an actual sentence following A (labelled `isNext`) and for the rest of the time it is a random sequence from the corpus (labelled `notNext`). `[CLS]` is used for this binary prediction task during pretraining. Therefore, during fine-tuning, the

embedding of `[CLS]` can be treated as the sequence embedding.

### A Lite BERT for Self-supervised Learning of Language Representations (ALBERT)

ALBERT [13] applies two methods for parameter reduction, **factorized embedding parameterization** and **cross-layer parameter sharing**, to the backbone of the BERT model and achieved significantly better performance on several NLP tasks while keeping the number of parameters 18x smaller and training time 1.7x faster.

**Factorized embedding parameterization** decouples the token embedding size $E$ and the hidden layer size $H$, i.e $E \neq H$. While $E$ is the size of *context-independent* embedding, $H$ is the size of *context-dependent* embedding, thus tying them together risks scaling up numbers of unnecessary parameters or preventing $H$ from being arbitrarily deep. As Liu et al. [15] pointed out, the power of a BERT-like representation is the ability to utilize context to influence *context-dependent* representation, which dictates that $H \gg E$. In ALBERT, instead of projecting the one-hot encoding of vocabulary directly into the hidden space, it was first projected into a lower dimension space of size $E$, and then projected onto the hidden space of size $H$. This effectively reduces the number of parameters from $O(V \times H)$ to $O(V \times E + E \times H)$, in which $V$ is the vocabulary size.

**Cross-layer parameter sharing** in ALBERT is the default for all parameters (feed forward and attention) across layers. This technique is thought to help stabilize the network while acting as a form of regularization to help with generalization.

Besides, for the second task of pretraining, instead of training on NSP as BERT does, ALBERT implements **inter-sentence coherence loss** or **sentence-order prediction loss** (SOP). Many studies have shown that NSP's impact might be unreliable and have since decided to remove it from pretraining [29]. Specifically, NSP formulates topic prediction and coherence prediction in a single task (predicting the next sentence). However, topic prediction seems simpler compared to coherence prediction, and might have been captured in the Masked LM. SOP, on the other hand, focuses on modeling inter-sentence coherence. SOP positive examples are pairs of con-

secutive sentences in the corpus, as in NSP, and negative examples are those pairs but with their orders swapped. This forces the model to learn more granular distinctions about the level of coherence between sentences.

**Fine tune Language Models for Question Answering**

In the Question Answering task, the input question and context are represented as a pair of sequences, with question using `A` embedding and answer using `B` embedding. As shown in Figure 3-2, at fine-tuning, a start vector $S$ and an end vector $E$ are introduced in the output layer. The probability of word $i$ with contextualized embedding $T_i$ being the start of the answer is computed as the dot product between $S$ and $T_i$ followed by softmax over all words in the passage: $P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$. A similar formula is used for the end of the answer span. The probability of an answer span from $i$ to $j$ ($j \geq i$) is therefore $\propto e^{S \cdot T_i} e^{S \cdot T_j} = e^{S \cdot T_i + S \cdot T_j}$. Thus, the score of a candidate answer span from $i$ to $j$ is defined as $S \cdot T_i + S \cdot T_j$. The training objective is the sum of the log-likelihoods of the correct start and end positions. Fine-tuned hyperparameters are (1) epochs, learning rate, and batch size (for QA tasks), and (2) max sequence length, and window size (for sliding window).

To represent questions that do not have an explicit answer in the documents, the answer span is extended to include `[CLS]`, with embedding $C$. For prediction, the score of a null answer (an answer that starts and ends at the `[CLS]` token) is $s_{null} = S \cdot C + E \cdot C$ and the best scores of non-null answers are $\hat{s_{i,j}} = \max_{j \geq i} S \cdot T_i + E \cdot T_j$. The non-null answer is chosen when $\hat{s_{i,j}} > s_{null} + \tau$ where $\tau$ is the threshold of non-null differences (a hyperparameter that will be finetuned with the dev set).

As in Section 3.1, a sliding window approach divides the passage into many overlapping text spans. As the result, for a word $i'$ that appears in the overlapping part of two contexts, $T'_i$ is the embedding of $i'$ with maximal context. Context of a word or token is defined by a score function that balances both left and right context. As the result, $i'$ with maximal context is approximately in the middle of the span. For example, for the sentence `the man went to the store and bought a gallon of milk`, maximal length of 6 and a sliding window of size 3 gives three text spans (1) `the`

`man went to the store`, (2) `to the store and bought a`, and (3) `and bought a gallon of milk`, the embedding of the word `store` is the version of `store` in the second text span. Implementation of this approach follows BertForQuestionAnswering and examples in the Huggingface Transformer library [27].

### 3.2.4 Evaluation

The development set (used for fine-tuning) and test set are evaluated on Exact Match (EM) and F1 metrics. EM is a score in the $[0, 1]$ range, which represents the proportion of predicted answers that exactly matched the true answers. Let $\hat{a}$ be the predicted answer and $a$ be the true answer:

$$EM = \frac{\sum_{i=1}^{n} \mathbb{1}(\hat{a} \equiv a)}{n}$$

Since EM is the mean of a sample of 0s and 1s, its confidence interval can be calculated with a binomial distribution of size $n$ and probability of success $EM$.

F1 for a pair of $(\hat{a}, a)$ is a score in the $[0, 1]$ range and measures the proportion of words overlapping between the predicted answer $\hat{a}$ and the true answer $a$. For a validation dataset, the average F1 score is reported.

## 3.3 Study of Attention Head

As discussed in Section 3.2.3, both BERT and ALBERT are made of Transformer encoders. Transformer is a building block that consists of multiple attention heads in each layer. Given an input sequence of vectors $x = [x_1, ..., x_n]$ (where $x_i$ can be seen as a vector of token embeddings), an attention head transforms each $x_i$ into query, key, and value vectors $q_i, k_i, v_i$ through separate linear transformations. Between all pairs of token embeddings $(x_i, x_j)$, the attention head computes an attention weight as the softmax-normalized dot product between query and key vector $\alpha_{i,j} = \frac{exp(q_i \cdot k_j)}{\sum_{m=1}^{n} exp(q_i \cdot k_m)}$. The output of the attention head is calculated as the weighted sum of the value vectors $o_i = \sum_{j=1}^{n} \alpha_{i,j} v_j$. The attention weights $\alpha_{i,j}$ thus define how much attention

token $j$ contributes to the representation of token $i$ in the next iterations.

This work studies attention heads in the context of a QA task, which makes it different from other studies about attention head attention mechanisms. Compared to the setting that uses consecutive paragraphs, a QA pair of sequences exhibits not only topic similarity and coherence but also a question-answer relationship. Here, a query-context pair is denoted as `< [CLS] query` $[SEP]_1$ `context` $[SEP]_2$ `>`.

In this work, I seek to test the hypothesis that BERT and ALBERT utilize basic information of cloze queries, such as cloze position and question mark, similar to human readers. I hypothesize that BERT and ALBERT pay significant attention to (`[CLS]`, and `[MASK]` (cloze position), and '?' tokens in the query .

Randomly selecting 100 documents from the CliCR development set, about 1500 context-answer pairs are formed using the sliding window technique described in Section 3.1. The max sequence length is 384 and document stride is 128 (tuned for the QA task with dev set—Table A.1). Here, attention head $h$ in layer $l$ is abbreviated as $At(h, l)$. From this dataset, an average attention weight is calculated for each $At(h, l)$. This average attention weight shows how much attention $At(h, l)$ contributes to certain tokens. Attention dispersion at $At(h, l)$ is measured through average entropy, where entropy of the attention distribution at position $x_i$ is defined as:

$$Entropy_{\alpha}(x_i) = -\sum_{j=1}^{n} \alpha_{i,j} log(\alpha_{i,j})$$

While the quantitative analysis is done at the token level, visualization of attention patterns is done at the word level. The token-token attention map is converted to a word-word attention map by a two-step process. First, for attention *to* a split-up word, the new attention weight is the sum of attention weights to its tokens. Then, for attention *from* a split-up word, the new attention weight is the mean of attention weights over its tokens. These transformations preserve the property that attention weights *from* a token sum to 1.

# Chapter 4

# Results and Discussion

## 4.1 Question Answering on CliCR

*This section presents the results from the QA experiments proposed in Chapter 3, and discusses important key takeways. Throughout this section, CliCR* `X` *denotes the dataset formed by replacing @placeholder in the original CliCR dataset with* `X`*. The exception is maskedLM, in which a BioBERT model trained on CliCR* `[MASK]` *filled Wh- words in @placeholder (or* `[MASK]` *position). Another BioBERT model is then trained on this new dataset and the results are reported. Similarly, SQuAD* `[MASK]` *denotes the SQuAD dataset with question words masked out. HasAnsExact denotes the subset of the queries for which the answers are verbatim in the passages. NoAns denotes the subset of the queries for which answers are not explicitly mentioned in the passages.*

Since the answers in CliCR are entities, during the first set of experiments, *@placeholder* in the query is filled with `What`. Some questions, such as "What from amniotic band disruption is a possibility ?", turn out to be quite natural. Table 4.1 shows the result from my proposed models on the full training data of CliCR `What`, as well as the human benchmark scores from Sŭster et al [24]. Overall, biobert_v1.1_pubmed has the highest EM and F1 scores on both fine-tuned development set and test set, although its 95% CI of EM ([0.5344-0.559] for dev set and [0.5318-0.5550] for test set) are not significantly different than other language models, bert-base-case and

| Model | Dev ($n = 6391$) | | Test ($n = 7184$) | |
|---|---|---|---|---|
| | EM | F1 | EM | F1 |
| Attention Sum (baseline) | | | 10.4 | 23.1 |
| Gated Attention | 22.8 | 33.1 | 22.2 | 32.2 |
| bert-base-case | 54.04 | 57.49 | 53.22 | 56.60 |
| biobert_v1.1_pubmed | 54.67 | 58.26 | 54.34 | 57.75 |
| albert-base-v2 | 54.62 | 57.06 | 54.36 | 56.68 |
| xlnet-large-case | 17.96 | 23.41 | - | - |
| *human expert* | - | - | *35* | *53.7* |
| *human novice* | - | - | *31* | *45.1* |

Table 4.1: Question answering results on full development and test sets. The human scores (in *italic*) are from Sŭster et al [24]. The models are trained and evaluated with CliCR `What` in these experiments.

albert-base-v2. The better performance of biobert_v1.1_pubmed can be explained by the fact that it has been pretrained on 1M PubMed documents, and thus has acquired biomedical vocabulary and domain structure. Despite not being pretrained on medical documents and using many fewer parameters (11M parameters versus 110M parameters for BERT), albert-base-v2 has very close EM and F1 scores on both the development and test set compared to biobert_v1.1_pubmed. This performance of albert-base-v2 might be explained by the stability of the attention head, which will be explored in a later section and in Appendix C.

Table 4.1 also shows that the human benchmark scores are significantly lower than the language models'. That is largely due to the openness of the gap-filling question type (more than one answer is possible and some answers are not explicitly mentioned in the passages). Moreover, due to the automated construction of the dataset, some queries are left unanswerable by the passage. The human readers might have a hard time determining if the correct answer is contained in the passage, and if so, where it appears in the passage. On the other hand, language models, especially biobert_v1.1_pubmed, which are trained on 1M medical documents, might be able to capture more information, especially as compared to a novice human reader.

Figure B-5a shows the EM and F1 scores of biobert_v1.1_pubmed on CliCR when trained on 50%, 75%, and 100% of the train set. From each passage of the

| Model | HasAnsExact ($n = 3844$) | | NoAns ($n = 2547$) | |
|---|---|---|---|---|
| | EM | F1 | EM | F1 |
| bert-base-case | 37.70 | 43.42 | 78.72 | 78.72 |
| biobert_v1.1_pubmed | 39.15 | 45.12 | 78.09 | 78.09 |
| albert-base-v2 | 33.69 | 37.73 | 86.21 | 86.21 |

Table 4.2: EM and F1 scores of three language models on queries with or without answers in the development set.

train set, 50% or 75% of the queries are randomly sampled without replacement to create a smaller train set of size 50% or 75% of the original train set. The sampling process is repeated three times. The EM and F1 scores increase significantly ($p < 0.05$ one-sided less than in Fisher Exact Test) as the amount of training data increases to 100%. This shows that biobert_v1.1_pubmed relies heavily on the train set of CliCR during fine-tuning. On the other hand, Figure B-5b shows that, for a similar sampling process, performances of bert-base-case trained on 50%, 75% or full train set of SQuADv2.0 don't increase significantly as the amount of training data increases. This suggests that CliCR poses a more difficult QA task, compared to SQuADv2.0. The difficulty of CliCR might also explain why human and machine readers' F1 scores on this dataset are well below the theoretical benchmark of 100%.

The performance of three language models on CliCR in Table 4.1 are further stratified on HasAnsExact and NoAns subsets of the development set. The result is shown in Table 4.2. All three models have significantly higher EM and F1 scores for NoAns queries, as compared with HasAnsExact queries. Figure B-3 shows that both BERT and ALBERT tend to predict simple, short answers (Figure B-4) for answerable queries. These behaviors suggest that when facing hard answerable queries (and given the choice to deem those queries unanswerable), language models tend to predict the null answer.

The task to determine if an answer can be found in the passage (and if so, where) is challenging not only to human readers, but also to machine readers. Table 4.3 shows EM and F1 scores evaluated on only the HasAnsExact subset of the development set on two experiment settings, (1) train on full training set and (2) train on only the

| Model | Full ($n = 91344$) | | HasAnsExact ($n = 56267$) | |
|---|---|---|---|---|
| | EM | F1 | EM | F1 |
| bert-base-case | 37.70 | 43.42 | 47.42 | 57.64 |
| biobert_v1.1_pubmed | 39.15 | 45.12 | 52.03 | 62.73 |
| albert-base-v2 | 33.69 | 37.73 | 43.44 | 50.52 |

Table 4.3: Question answering results evaluated on **queries with verbatim answer in the development set** ($n = 3844$). Results are reported when trained on the full dataset (include queries with no explicit answer) and on only queries with verbatim answer. In these experiments, *@placeholder* in the query is filled with "What".

HasAnsExact subset of the training set. Across all three experiments, the EM and F1 score are significantly higher in the second setting. This observation is consistent in SQuADv1.1 and SQuADv2.0. As discussed in Chapter 2, SQuADv1.0 is exactly the HasAnsExact subset of SQuADv2.0. Table A.2 shows that SQuADv2.0 has significantly lower F1 in HasAnsExact than SQuADv1.0 ($77.59 < 87.66$) for the same model, bert-base-case. These observations suggest that eliminating the uncertainty of whether a query is answerable boosts the performance of machine comprehension on answerable queries.

## 4.2 Basic elements of cloze queries

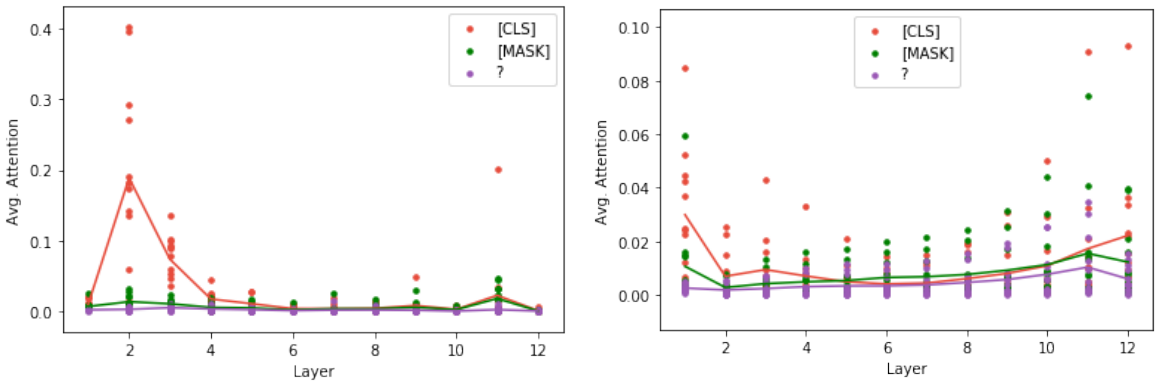
Next, I study the effects of Wh- words and the question mark token '?' on the QA task of CliCR. Table 4.4 shows the EM and F1 scores on the full development set and its HasAnsExact subset for varying formulations of queries. From Table 4.4, the "Wh- word" has little to no impact on the EM and F1 scores, as changing those words doesn't change EM and F1 scores significantly. The same is observed when masking out Wh- words in the SQuAD datasets (Figure A.2). However, omitting '?' in queries filled with `[MASK]` significantly lowers EM score for HasAnsExact subset of the development set (mean EM for `[MASK]` is significantly greater than mean EM for `[MASK]` + omit '?' with p-value of 0.034 by Fisher Exact Test). Leaving the gap in the query (denoted by `[MASK]`) results in the highest EM and F1 scores in

| Question word | Full ($n = 6391$) | | HasAnsExact ($n = 3844$) | |
|---|---|---|---|---|
| | EM | F1 | EM | F1 |
| `[MASK]` | 55.20 | 59.08 | 40.32 | 46.76 |
| what | 54.67 | 58.26 | 39.15 | 45.12 |
| why | 55.23 | 58.63 | 38.63 | 44.28 |
| which | 55.01 | 58.76 | 40.14 | 46.36 |
| how | 55.25 | 58.84 | 39.75 | 45.71 |
| *maskedLM* | 55.59 | 58.85 | 38.21 | 43.62 |
| what + omit '?' | 54.81 | 58.28 | 38.40 | 44.20 |
| `[MASK]` + omit '?' | 55.09 | 58.26 | 38.27 | 43.53 |

Table 4.4: Question answering results of biobert_v1.1_pubmed different formulation of queries evaluated on (1) the whole development set and (2) part of the development set with verbatim answer. The *@placeholder* are replaced with `[MASK]`, or other question words. In the *maskedLM* experiment, a `biobert` model trained on CliCR `[MASK]` is used to fill Wh- words into the cloze position, which can then be used to train another BertForQuestionAnswering model.

HasAnsExact subset of the development set and highest F1 overall. This can be explained by the fact that language models, especially BERT and ALBERT, have been pretrained for MaskedLM tasks. Moreover, forcing Wh- words in the blank makes most queries unnatural (Wh- words end up at the end of the sentences in some cases). The language models might not have seen these unnatural queries during pretraining, and thus, might not attribute their attention weights accordingly.

To study whether Transformer models focus on basic elements of a cloze query, such as cloze position *@placeholder* and question mark, I studied the attention patterns to `[CLS]` (determine if a querry can be answered), `[MASK]` (represents cloze position), and '?'. Figure 4-1a shows that BERT's attention heads at layer 2 and 3 pay a significant amount of attention to `[CLS]` (about 20% of their attention on average, with some attention heads paying as much as 40% of their attention). Although quite a lot of attention from BERT attention heads focus on `[CLS]` in layer 2, some heads pay significantly more attention than the others and the attention to `[CLS]` is much broader. This is shown in the heatmap of average attention in Figure 4-2a and the high entropy of attention distribution on `[CLS]` in Figure 4-3a. The attention on `[CLS]` from attention heads becomes broader and less concentrated in the deeper



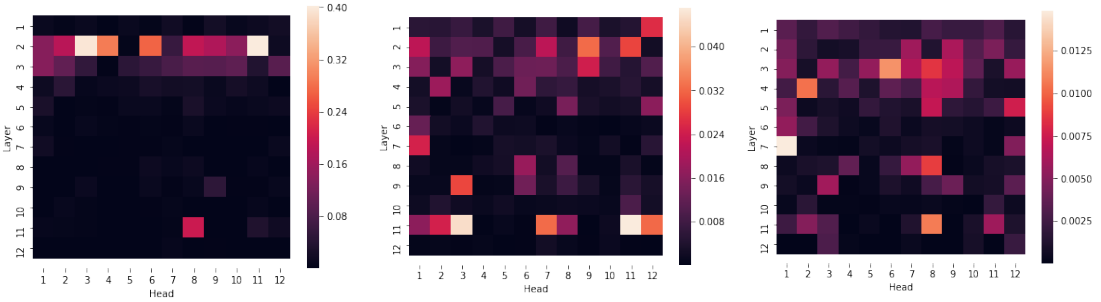

(a) BERT (b) ALBERT

Figure 4-1: Average attention of attention heads across layers in biobert_v1.1_pubmed and albert-base-v2 model trained on CliCR `[MASK]`. The solid line goes through the average of attention heads in each layer.

layers. This might suggest that `[CLS]` can act as a sequence embedding. This special token takes a lot of attention from other tokens at the beginning to improve its embedding. In the deeper layer, `[CLS]` receives less attention, as at this time it might propagate the earlier information to other parts of the sequence.

On the contrary, ALBERT attention heads don't pay as much attention to `[CLS]`. Attention heads of ALBERT experience somewhat similar patterns in early layers as BERT's, with attention heads in layer 1 (as shown in Figure 4-4a) paying more attention to `[CLS]` than in deeper layers. However, the average attention is much lower (only a little more than 8% of attention at the highest, as seen in Figure 4-1b and Figure 4-4a). However, the attention on `[CLS]` in ALBERT is more focused, with low entropy of attention distribution as shown in Figure 4-3c. Last but not least, Figure 4-4a shows that Head 6 pays more attention to `[CLS]` as the layer gets deeper. This different pattern in ALBERT, as compared to BERT, might be a result of their different second pretrained task that involves `[CLS]` (Sections 3.2.3 and 3.2.3). This might also explain the higher performance of ALBERT on NoAns subset of the development set, as compared to BERT (Table 4.2), as `[CLS]`'s embedding determines if a null answer is predicted.

In general, attention heads in both BERT and ALBERT don't pay much attention to `[MASK]` (cloze position) or '?', as shown in Figure 4-1. The heatmaps of average

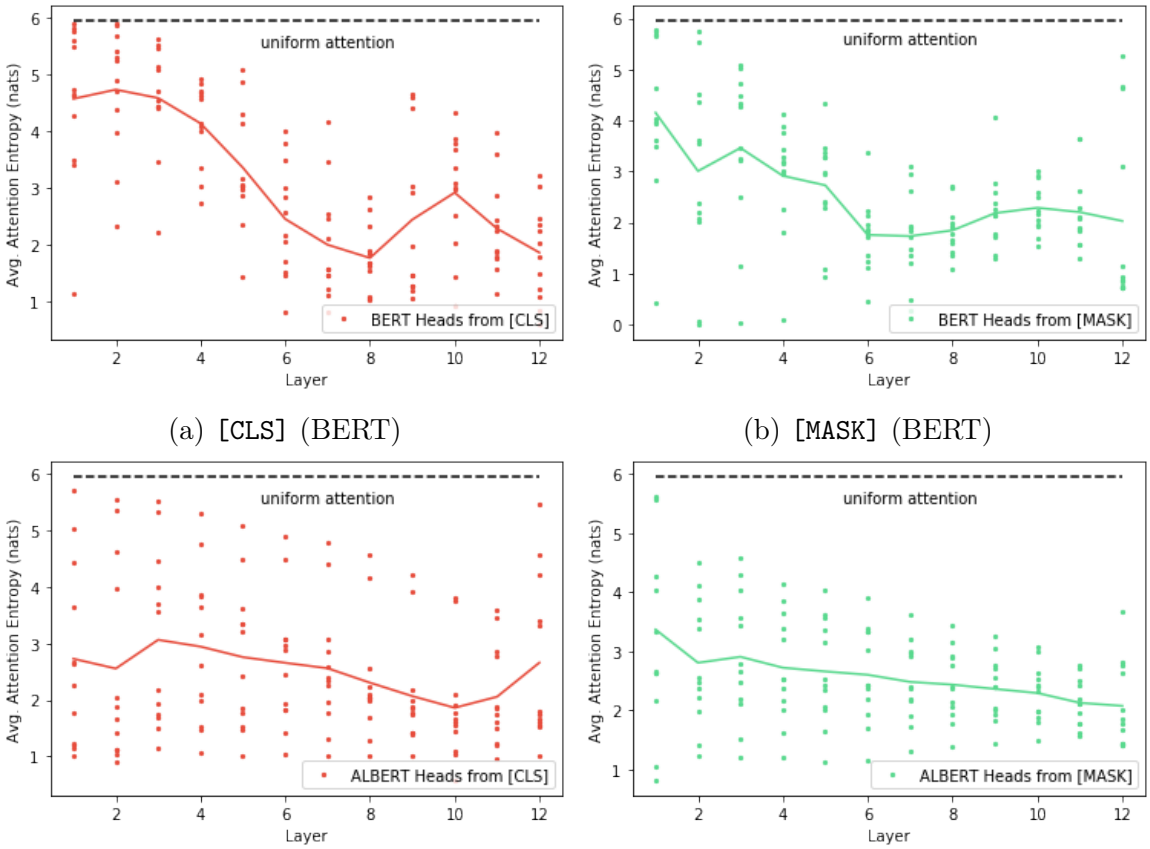



(a) [CLS] (b) [MASK] (c) '?'

Figure 4-2: Heatmap of average attention on cloze question elements of attention heads (by layers) in biobert_v1.1_pubmed model trained on CliCR [MASK]

attention in Figure 4-2 and 4-4 show that BERT and ALBERT attention heads don't have distinct attention patterns to these tokens in early layers. In deep layers in both BERT and ALBERT, there are a few attention heads with more attention focus on [MASK], marked by lower entropy of attention distribution as shown in Figure 4-3b and 4-3d. Figure 4-2b shows that At(3,11) and At(11,11) in BERT pay about 5% of their attention to [MASK]. Figure 4-4b shows that head 10 increases its attention to [MASK] in deeper layers, and spends as much as 7% of its attention on [MASK] in layer 11. However, whether this pattern is truly significant is not clear from this analysis.

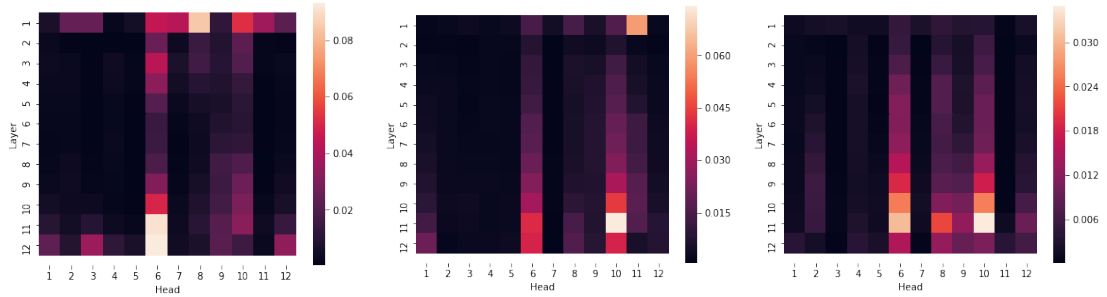
These observations support the results in Table 4.4, which suggest that [MASK] at cloze position and '?' might help with the performance of BERT/ALBERT for QA. However, this analysis fails to show that LMs pay significant attention to basic elements of questions, such as question words or cloze positions, and '?', the same way that human readers do.




(a) `[CLS]` (BERT) (b) `[MASK]` (BERT)

(c) `[CLS]` (ALBERT) (d) `[MASK]` (ALBERT)

Figure 4-3: Entropy of attention distribution of attention heads of BERT and ALBERT models on `[CLS]` and `[MASK]` tokens




(a) `[CLS]` (b) `[MASK]` (c) '?'

Figure 4-4: Heatmap of average attention on cloze question elements of attention heads (by layers) in albert-base-v2 model trained on CliCR `[MASK]`

# Chapter 5

# Conclusion and Future Work

## 5.1 Conclusion

My work on QA on CliCR has achieved state-of-the-art performance, with 55.2 exact match (EM) accuracy and 59.8 F1 score using the BioBert QA model ([14]), higher than the current best performer, gated-attention machine readers (EM=22.2, F1=32.2, [6]) and human expert readers (EM=35, F1=53.7, [24]). Further analysis of the approaches attempted in this work reveals the following key observations:

- Cloze queries can be modified systematically at the cloze position to apply LMs for QA. This approach leads to higher EM and F1 score than conventional methods for gap filling, such as Gated Attention Reader, as it benefits from transfer learning (pretraining) of LMs.

- CliCR poses a more difficult QA task, compared to SQuADv2.0. This can be explained by the domain of knowledge (medical domain versus general knowledge), the openness of answers (more than one possible answer and some answers not verbatim in the passage), type of queries (gap filling queries versus natural questions), and the length and complexity of documents. This can also explain the huge gap between the best human comprehension performance and theoretical bound of 100% for F1 score.

- The uncertainty of whether an exact-match answer can be found in the context

makes the QA task harder for LMs. While LMs perform well with simple, short answers, when facing a hard, answerable query and given the choice of a null answer, LMs tend to predict the null answer.

- Analysis of attention heads shows some interesting observations of attention patterns to basic elements of a cloze query (`[CLS]`, `[MASK]`, '?'). However, whether these patterns are significant is not clear from the observations, as well as from an ablation study of question words and '?' (Section 4.2). It suggests that LMs might not interpret questions based on basic linguistic elements, such as question words, cloze position, and question mark, the same way human readers do.

## 5.2 Future Work

This thesis provides the groundwork for future improvements and extension.

- In this work, xlnet-large-case has lower performance than other BERT and ALBERT models, despite its ability to handle longer contexts and its larger vocabulary. In future work, I want to understand why it is the case, which may also be due to a problem with my implementation.

- Future extension of Section 4.2 might seek to understand the role of question in LMs for QA. Questions might provide a more narrow context to improve the contextualized embeddings of the document. Another approach is to show that embedding of questions influences the embedding of the start and end vectors of answer span (Section 3.2.3).

- Lower performance of LMs on CliCR might be due to the length of the document. Future work might seek to narrow the context for each query, so as to evaluate the impact of length of document on LMs for QA task.

# Appendix A

# Tables

| Models | Model Configs | Training configs |
|---|---|---|
| GA | #hidden layers: 128<br>Embedding size: 200<br>Gradient clip threshold: 10<br>Gradient step: 1<br>Dropout: 0.2 | Batch size: 32<br>#epochs: 10<br>Learning rate: 0.0005 |
| BERT<br>(`bert-base-case`,<br>`biobert_v1.1_pubmed`) | #hidden layers: 12<br>#attention heads: 12<br>Embedding size: 768<br>Hidden size: 768<br>Feed forward size: 3072<br>Dropout (all components): 0.1 | Max sequence length: 384<br>Max query length: 64<br>Document stride: 128<br>Batch size (per GPU): 12<br>#epochs: 2<br>Attention dropout: 0.1<br>Learning rate: 3E-5 |
| ALBERT<br>(`albert-base-v2`) | #hidden layers: 12<br>#attention heads: 12<br>Embedding size: 128<br>Hidden size: 768<br>Hidden dropout: 0.0<br>Attention dropout: 0.0 | Max sequence length: 384<br>Max query length: 64<br>Document stride: 128<br>Batch size (per GPU): 12<br>#epochs: 2<br>Learning rate: 3E-5 |
| XLNET<br>(xlnet-large-case) | #hidden layers: 24<br>#attention heads: 16<br>Embedding size: 1024<br>Hidden size: 1024<br>Feed forward size: 4096<br>Dropout (all components): 0.1 | Max sequence length: 384<br>Max query length: 64<br>Document stride: 128<br>Batch size (per GPU): 2<br>#epochs: 3<br>Learning rate: 1E-5 |

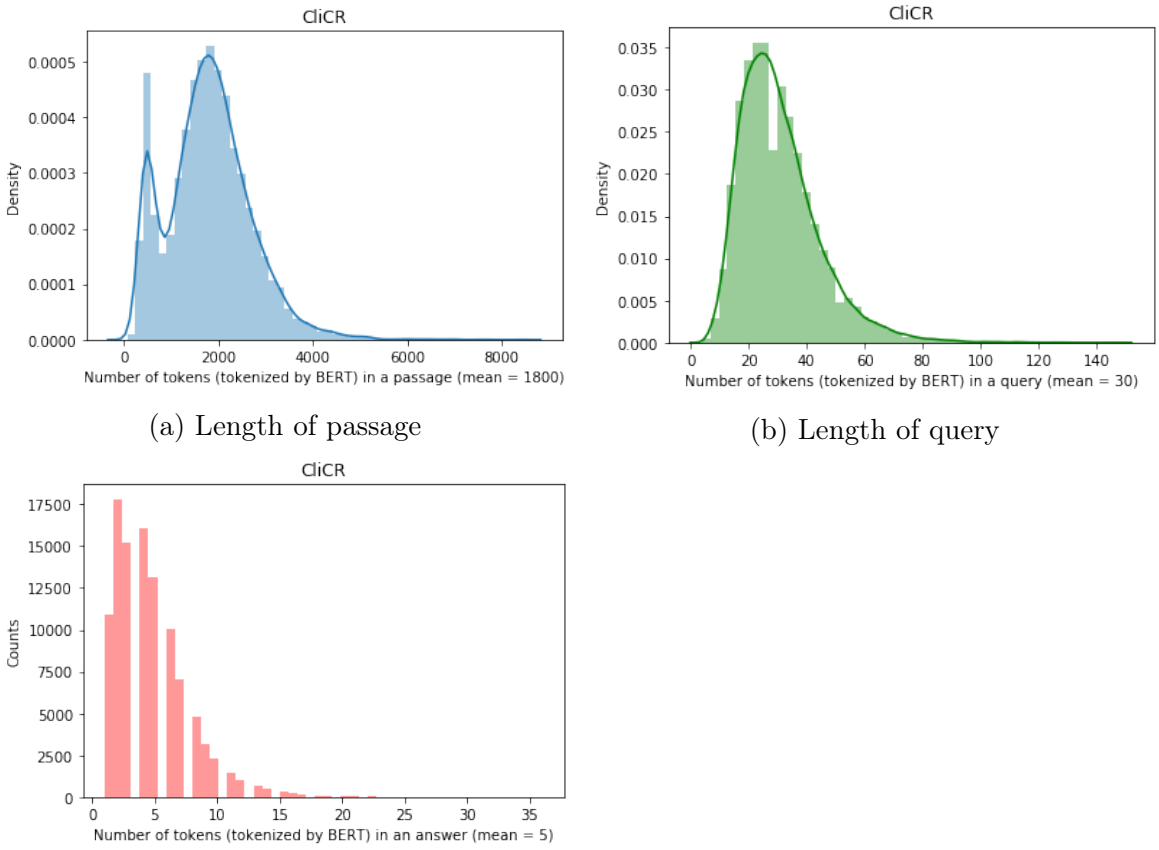
Table A.1: Configurations of QA task models

| Dataset | Full | | HasAnsExact | |
|---|---|---|---|---|
| | EM | F1 | EM | F1 |
| SQuADv1.1 | 79.57 | 87.67 | 79.57 | 87.66 |
| SQuADv1.1 `[MASK]` | 78.29 | 86.36 | 78.28 | 86.36 |
| SQuADv2.0 | 72.09 | 75.53 | 70.70 | 77.59 |
| SQuADv2.0 `[MASK]` | 71.43 | 74.79 | 69.20 | 75.92 |

Table A.2: SQuADv1.1 and SQuADv2.0 results reproduced with bert-base-case model on two 12G NVIDIA GPU. The training configuration is reported in Table A.1. EM and F1 are evaluated on full and HasAnsExact development set. All queries in SQuADv1.1 have answers. SQuADv1.1 is exactly the HasAnsExact subset of SQuADv2.0.

# Appendix B

# Figures

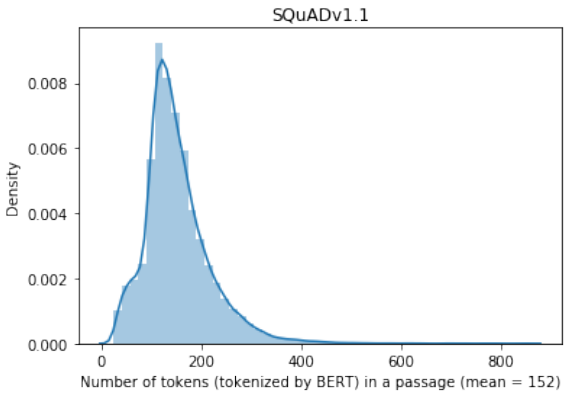



(a) Length of passage

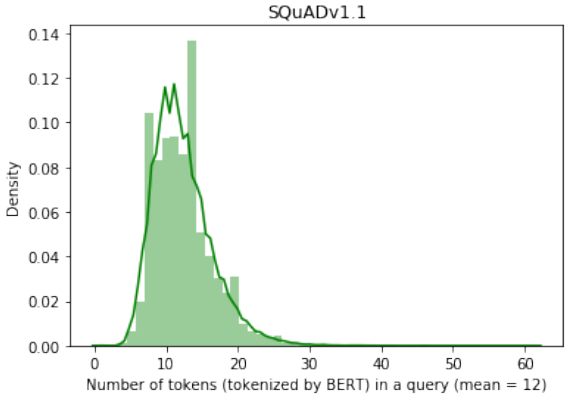
(b) Length of query

(c) Length of answer

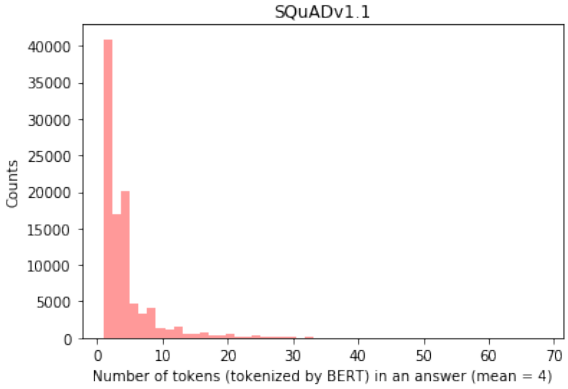
Figure B-1: Distribution of length (in tokens) of passages, queries, and answers in CliCR.




(a) Length of passage

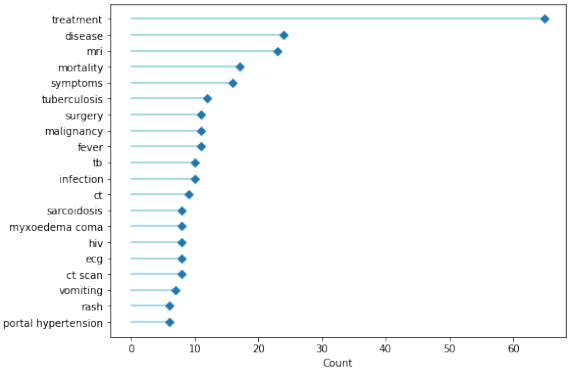



(b) Length of query

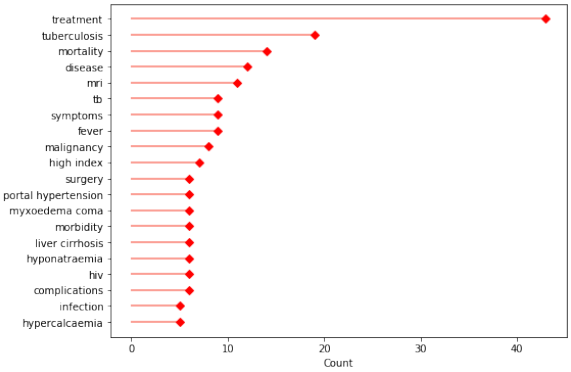



(c) Length of answer

Figure B-2: Distribution of length (in tokens) of passages, queries, and answers in SQuAD.

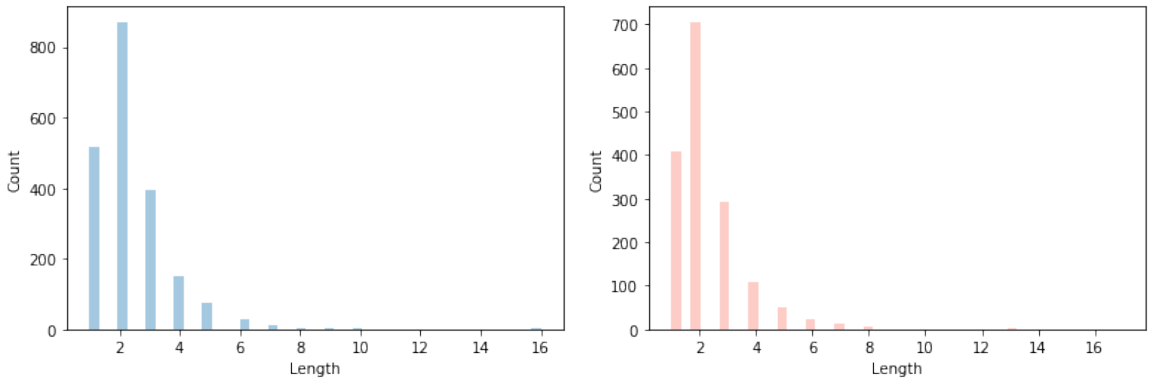



(a) biobert_v1.1_pubmed



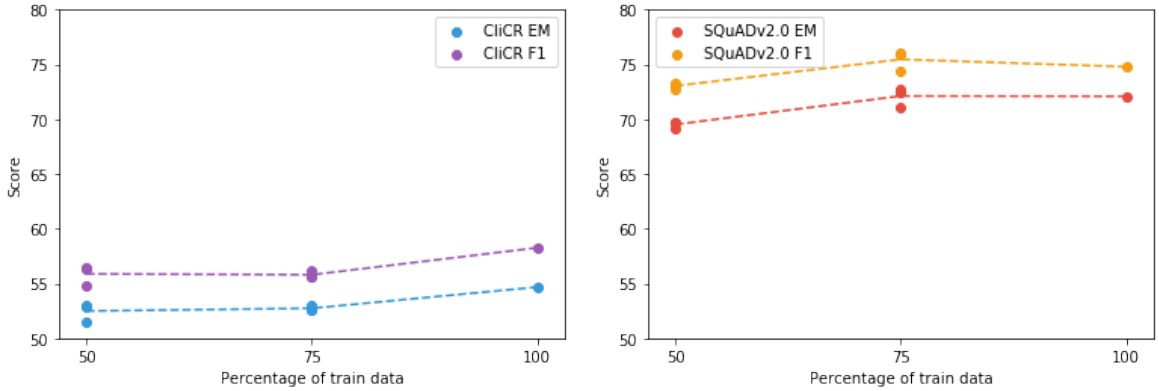

(b) albert-base-v2

Figure B-3: Top 20 most common answers as predicted by BioBERT and ALBERT models




(a) biobert_v1.1_pubmed (b) albert-base-v2

Figure B-4: Answer length distribution of BioBERT and ALBERT models




(a) CliCR (b) SQuADv2.0

Figure B-5: EM and F1 scores of CLiCR (trained with biobert_v1.1_pubmed) and SQuADv2.0 (trained with bert-base-case) when training on 50%, 75% and 100% of the train set. Each data point corresponds to score of a subset. The dotted line goes through the mean.

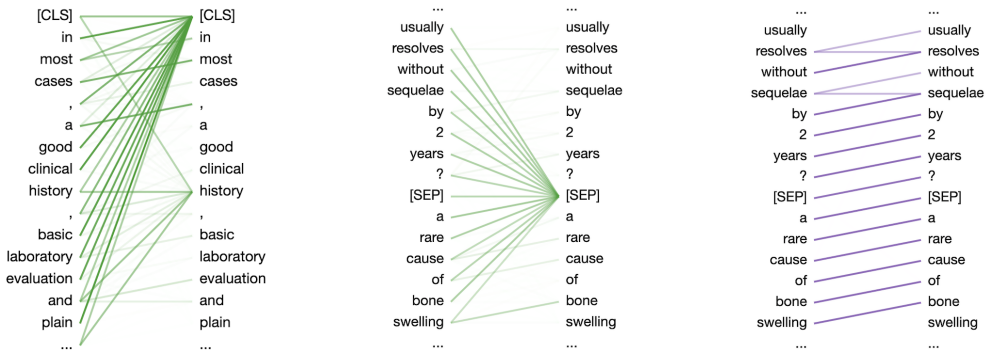
# Appendix C

# Study of Attention Head

*This appendix presents some observations about other attention mechanisms exhibited by attention heads in the BERT and ALBERT models, in addition to those discussed in Section 4.2.*

In this exploratory section, I focus on the attention weights on special tokens, `[CLS]`, `[SEP]`, `[MASK]` and '?'. Before looking at specific attention weights and distribution, Figure C-1 shows some surface-level attention patterns of BERT's attention heads.

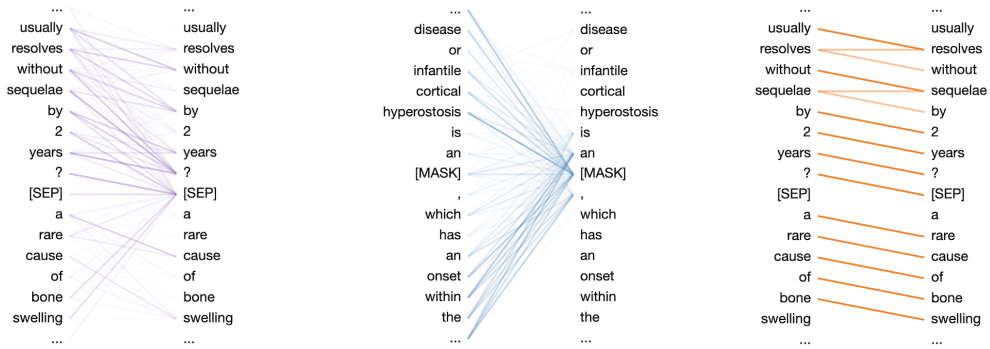
Figure C-2a show that the majority of attention heads in every layer in BERT pays more than 50% of attention to `[SEP]`, suggesting a strong attention to this special token. One possible hypothesis might be that `[SEP]` aggregates segment-level information, which can then propagate through other heads. However, further exploration reveals that this hypothesis might not be true. Figure C-2c shows that there are no differences between the two `[SEP]` tokens, despite the fact that $[SEP]_1$ is closer to the question and $[SEP]_2$ is closer to the context. If `[SEP]` tokens were meant to capture segment-level information, the attention level at these two `[SEP]` tokens would be somewhat different. In addition to having similar average attention in both `[SEP]` tokens, Figure C-4 shows that in both BERT and ALBERT models, the majority of attention to `[SEP]` tokens are from `[SEP]` tokens in the majority of attention heads. This suggests that information to `[SEP]` would not be able to propagate to other tokens. An exception is attention heads in layer 12 in BERT




(a) Head 3 Layer 2 - attention to `[CLS]`

(b) Head 3 Layer 7 - attention to `[SEP]`

(c) Head 5 Layer 2 - attention to previous token

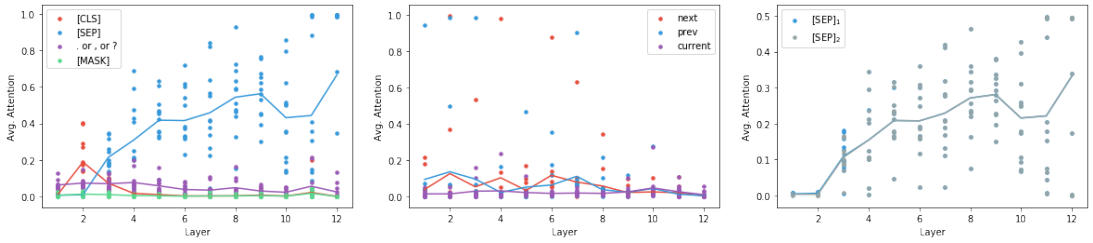



(d) Head 5 Layer 3 - attention to '?'

(e) Head 11 Layer 11 - attention to `[MASK]`

(f) Head 12 Layer 2 - attention to next token

Figure C-1: Examples of heads with attention mechanism described in Section 4.2. The darkness of lines indicates the strength of the attention weight.

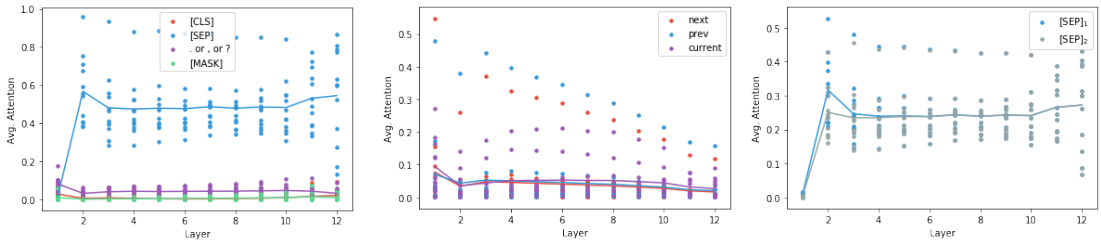



(a) Special tokens and punctuations

(b) Self-attention, next and previous tokens

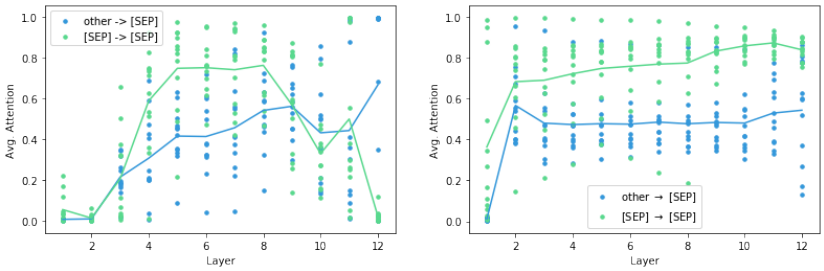
(c) Two different positions of `[SEP]`

Figure C-2: Average attention of attention heads across layers in `biobert_v1.1_pubmed` model trained on CliCR `[MASK]`. The solid line goes through the average of attention heads in each layer.




(a) Special tokens and punctuations (b) Self-attention, next and previous tokens (c) Two different positions of `[SEP]`

Figure C-3: Average attention of attention heads across layers in `albert_base_v2` model trained on CliCR `[MASK]`. The solid line goes through the average of attention heads in each layer.

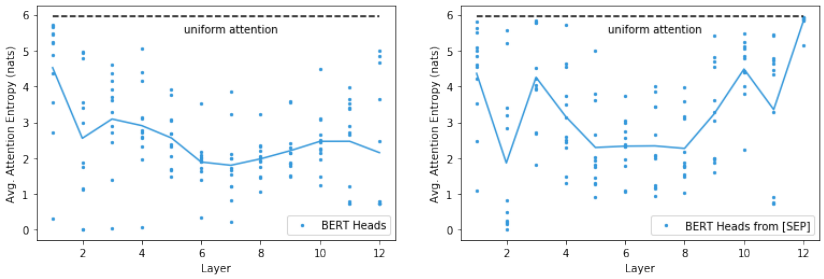



(a) BERT (b) ALBERT

Figure C-4: Attention to `[SEP]` from `[SEP]` and other tokens in biobert_v1.1_pubmed and albert-base-v2 models.

models, which have other tokens paying more attention to `[SEP]` tokens than `[SEP]` tokens themselves. However, Figure C-5b shows that this might be a result of a broad attention to `[SEP]` tokens. At those attention heads in the deeper layer (layer 12), the entropy of the attention distribution at `[SEP]` almost equals the entropy of uniform attention. Together, these observations suggest that attention to `[SEP]` might be used as "no other options" when the attention head patterns are not applicable, agreeing with observations from Clark et al. [4]. These observations also hold in ALBERT's attention heads. However, thanks to cross-layer weight sharing, average attention weights (Figure C-3a) and the entropy of the attention distribution (Figure C-6b) at `[SEP]` tokens are more stable across layers in ALBERT, as compared to BERT.

Next, attention to relative positions is explored. Both BERT and ALBERT exhibits some patterns of attentions to previous and next tokens. Figure C-2b and Figure C-3b show that several attention heads in early layers (layer 1 through 4) pay

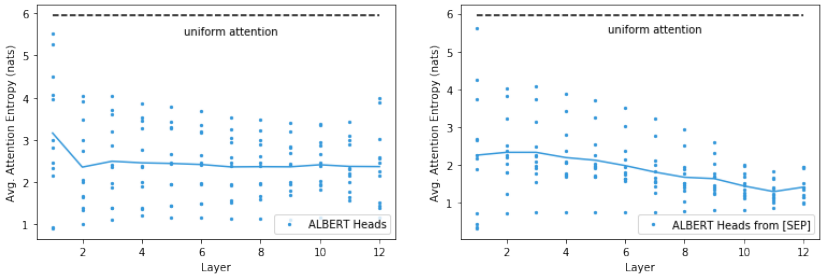



(a) All tokens (b) `[SEP]`

Figure C-5: Mean entropy of attention heads across layers in `biobert_v1.1_pubmed` model trained on CliCR `[MASK]`. The solid line goes through the average of attention heads in each layer.

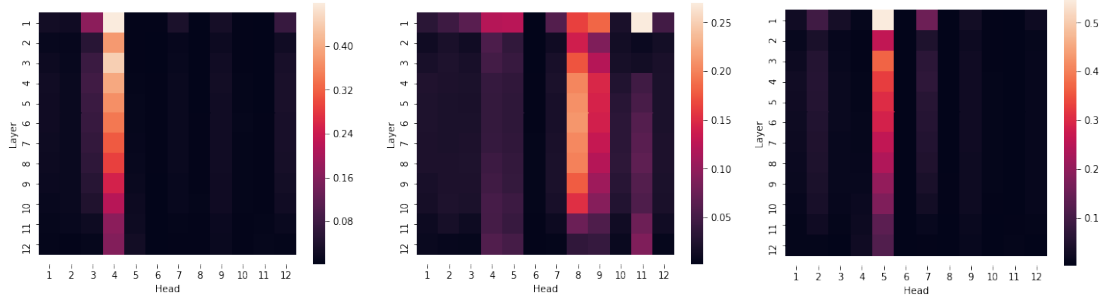
more than 50% of their attention to the previous and next tokens. BERT's attention heads seem to pay little to no attention to the current token, while ALBERT's attention heads pay significant attention to the current token. Compared to BERT's attention heads, ALBERT's attention heads exhibit a more stable and clearer pattern of attention to relative position. Figures C-7a to C-7c show clear patterns of attention heads at relative positions in the ALBERT model. Thanks to cross-layer parameter sharing in ALBERT, these patterns could persist across multiple layers.

Overall, many attention patterns (strong attention to `[SEP]` and to relation position) are similar between BERT and ALBERT. ALBERT attention heads seem to exhibit more focus, stable attention patterns (Figure C-6), as compared to BERT (Figure C-5). Thanks to cross-layer parameter sharing, ALBERT attention patterns are maintained in attention heads across layers (Figure C-7 and 4-4) while BERT's attention heads exhibit more sporadic attention patterns (Figure 4-2).




(a) All tokens (b) `[SEP]`

Figure C-6: Mean entropy of attention heads across layers in `albert_base_v2` model trained on CliCR `[MASK]`. The solid line goes through the average of attention heads in each layer.




(a) Previous token (b) Self (c) Next token

Figure C-7: Heatmap of average attention of attention heads across layers in `albert-base-v2` model trained on CliCR `[MASK]`.

# Bibliography


[1] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

[2] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[3] Stephen Chu and Branko Cesnik. Knowledge representation and retrieval using conceptual graphs and free text document self-organisation techniques. *International journal of medical informatics*, 62(2-3):121–133, 2001.

[4] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does BERT look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

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

[6] Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549*, 2016.

[7] Bhuwan Dhingra, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Linguistic knowledge as memory for recurrent neural networks. *arXiv preprint arXiv:1703.02620*, 2017.

[8] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.

[9] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

[10] Betsy L Humphreys and DA Lindberg. The umls project: making the conceptual connection between users and the information they need. *Bulletin of the Medical Library Association*, 81(2):170, 1993.

[11] Qiao Jin, Bhuwan Dhingra, William W Cohen, and Xinghua Lu. Probing biomedical embeddings from language models. *arXiv preprint arXiv:1904.02181*, 2019.

[12] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

[13] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

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

[15] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[16] Jan Pomikálek. justext. 2011.

[17] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

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

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

[20] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, 2013.

[21] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[22] Marco Antonio Calijorne Soares and Fernando Silva Parreiras. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University-Computer and Information Sciences*, 2018.

[23] Ergin Soysal, Jingqi Wang, Min Jiang, Yonghui Wu, Serguei Pakhomov, Hongfang Liu, and Hua Xu. Clamp–a toolkit for efficiently building customized clinical natural language processing pipelines. *Journal of the American Medical Informatics Association*, 25(3):331–336, 2017.

[24] Simon Šuster and Walter Daelemans. Clicr: A dataset of clinical case reports for machine reading comprehension. *arXiv preprint arXiv:1803.09720*, 2018.

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

[26] Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*, 2019.

[27] Thomas Wolf, L Debut, V Sanh, J Chaumond, C Delangue, A Moi, P Cistac, T Rault, R Louf, M Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv, abs/1910.03771*, 2019.

[28] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, 2015.

[29] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.