November 2019

ENABLING CONVERSATIONAL QUESTION ANSWERING USING A BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT) BASED MODEL

By

Munazza Zaib

A THESIS SUBMITTED TO MACQUARIE UNIVERSITY
FOR THE DEGREE OF MASTER OF RESEARCH
DEPARTMENT OF COMPUTING
NOVEMBER 2019

EXAMINER'S COPY

Typeset in LaTeX 2_ε.

Declaration

I certify that the work in this thesis entitled ENABLING CONVERSATIONAL QUESTION ANSWERING USING A BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT) BASED MODEL has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree to any other university or institution other than Macquarie University. I also certify that the thesis is an original piece of research and it has been written by me. Any help and assistance that I have received in my research work and the preparation of the thesis itself have been appropriately acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Munazza Zaib

Acknowledgements

First of all, I would like to express gratitude towards God Almighty for bestowing me with the knowledge, capacity, and opportunity to undertake this research study and to persevere and complete it satisfactorily. This success would not have been possible without His blessings.

I would like to offer my sincerest thanks to my advisor Prof. Michael Sheng and my co-supervisor Dr. Wei Emma Zhang for their consistent help during my Master's research. They have been there providing their ardent advice and direction consistently. They have given me significant guidance, motivation, and suggestions in my quest for knowledge. They have given me all the opportunity to choose my direction, while quietly and non-prominently guaranteeing that I remain on course and don't veer off from the core of my research. Without their capable direction, this thesis would not have been conceivable and I will interminably be thankful to them for their help.

I would like to express gratitude towards my family. The endowments of my parents, the everyday video calls of my siblings that kept things from getting dull or exhausting, and above all the genuine love and support of my better half, who made this long-distance relationship simple for me and furnished me with the significant serenity, have all made it possible for me to arrive at this phase in my life. I thank all of them for enduring me in troublesome minutes where I felt baffled and for prodding me on to pursue my dreams about getting this degree. This would not have been conceivable without their steady and unselfish love and backing is given to me consistently.

List of Publications

- **Munazza Zaib**, Quan Z. Sheng, Wei Emma Zhang, Dai Hoang Tran, *BERT-CoQAC: BERT-based Conversational Question Answering in Context*. (submitted to AAAI'2020)
- **Munazza Zaib**, Quan Z. Sheng, Wei Emma Zhang, *A Short Survey of Pre-trained Language Modeling for Conversational AI-A New Age in NLP*. (submitted to ACSW'2020)

Abstract

As one promising way to inquire about any particular information through a dialog with the bot, question answering dialog systems have gained increasing research interest recently. However, such systems often struggle when the dialog is carried out in multiple turns by the users to seek more information based on what they have already learned. Although several works have dealt with the issue of history modeling in multi-turn question answering, most have focused on either prepending the history questions and answers, employing complex attention mechanism or not capturing the entire context of the history conversation. To address this issue, we propose a BERT based **Co**nversational **Q**uestion **A**nswering in **C**ontext (BERT-CoQAC) that involves the seamless integration of the relevant context of conversational history into a BERT-based question answering model. This study proposes a framework that provides a dynamic history selection process and combines the embeddings of history answers along with the history questions to generate a complete input in order to capture the understanding of the current query more accurately. To test our approach, we performed extensive experiments and the results implies that having a proper history conversation modeling is necessary to achieve better results. We also studied the effect of not having a history selection mechanism to provide new insights into history conversation modeling.

Contents

List of Figures

List of Tables

1

Introduction

Having a virtual assistant with an intelligence level adequate enough to carry out a smooth dialogue has only existed in Sci-Fi movies and seemed like a dream for a long time. Over the past few years, conversational AI has caught a lot of attentions owing to its promising potential and powered many commercial aspects such as digital companion [67], and intelligent home controller like Google Home. The availability of "big data" combined with the advanced deep learning techniques has the potential to make this dream into a reality.The field of conversational AI can be divided in to three categories namely, *chat-oriented dialogue systems*, *goal-oriented dialogue systems*, and *question answering (QA) dialogue systems*. The former two have been very researched upon topics, resulting in a number of successful dialogue agents such as Apple Siri, Microsoft Cortana, and Amazon Alexa. However, QA dialogue systems are fairly new and still requires extensive research.

These question answering dialogue systems formulates the task of machine comprehension (MC), which provided a context paragraph, requires the machine to answer the current query. The main focus of this section is to study the task of machine comprehension, the advancements that has been made so far, the challenges pertaining to it, and propose our method to address the issues.

1.1 Machine Comprehension

The machine comprehension task dates back to 1970s where early machine comprehension systems, due to rule-based methods and absurdly small size of available datasets, did not achieve well thus, making it difficult to use them in practical applications. These systems saw their rise in 2015, which can be associated to two driving factors.

- One is the use of deep learning methods to capture the contextual information in machine comprehension tasks that outperforms the traditional rule-based models.
- The other factor that contributed in the progress is the availability of several large-scale datasets, such as SQuAD [45], MS MARCO [36], and CNN & DAILY MAIL [35], which made it possible to deal with the task of machine comprehension on neural architectures more efficiently and provide a test-bed for evaluating the performance of these models.

To make the machine comprehension task more similar to the real-world scenarios, a number of advanced challenges and orientations are introduced.

1.1.1 Multi-Passage Machine Reading Comprehension

In machine comprehension tasks, the related context is pre-determined, which is unlikely to the humans' way of question answering. Usually, people put forward a query first and then look into all the available knowledge they have to answer the query as accurately as possible. To address this research gap, Chen et al. [4] took machine comprehension to a whole new level by introducing the concept of machine reading comprehension form multiple passages, which did not restrict the answering of queries to only one passage unlike traditional approaches. This development can be used to deal with the question answering task based on open domain from massive unstructured corpora.

1.1.2 Knowledge-based Machine Comprehension

In a general scenario of machine comprehension task, the model is required to answer a query using the implicit knowledge in the provided context. Several datasets [47, 45] choose passages from selected corpora to avoid introducing external knowledge to the model. However, when compared to the real-world question answering process, these human-generated questions are very general. Human-reading comprehension can make use of the external knowledge to answer the query when its answer cannot be find using the provided context. The role of this extraneous knowledge is so critical that it is considered as the most sought after research area to fill the void between human-reading comprehension and

machine comprehension. This resulted in research community's growing interest in introducing world knowledge to machine comprehension and the concept of knowledge-based machine comprehension is introduced.

1.1.3 Unanswerable Questions

There is an inherent assumption when it comes to solving machine comprehension tasks that the given paragraphs always contains the right answers. However, this is in contrast with the real-world applications. The scope of information provided in the source passages is very particular and narrow. Then, some questions necessarily have no answers according to the information provided in the given passage. This challenge presented a new research area, *detection of unanswerable questions*, for the researchers and as a result several datasets are introduced to tackle the problem. One such dataset with 50,000 unanswerable questions is SQuAD [45].

1.1.4 Conversational Question Answering

In machine comprehension task, the idea is to answer the questions from the provided context and these questions are segregated from each other. In real-life scenario, however, people usually acquire knowledge by asking a series of questions and each new question might have some relation to the previously asked questions. This process is carried out continuously until a satisfactory amount of knowledge is gained and is termed as multi-turn question answering. After introducing the element of multi-turn question answering into machine comprehension , the Conversational Question Answering (CQA) or Conversational Machine Comprehension (CMC) has caught huge attentions from both industry and academia.

With its appearance, many researchers try to induce a conversational touch to satisfy the requirements for the task of CQA by introducing a source passage and a series of inter-related questions. Reddy et al. [46] release Conversational Question Answering (CoQA) dataset consisting of 8000 conversation passages from seven different domains. The seeker asks a question on the basis of given passage and the answer provider provides the answer, which mimics the conversation process that takes place between the two people when reading the given context. The answers are usually free-form which requires more context reasoning. Another dataset, QuAC, is introduced recently by Choi et al. [11] for question answering in context. QuAC urges user to participate more in information seeking dialog. Unlike CoQA, the information seeker has access only to the title of the paragraph and can pose free-form questions to learn about the hidden text of Wikipedia paragraph. The answer provider answers the question and identifies whether the seeker can ask question to follow-up or not.

The cloze test task was extended by Ma et al. [31] to suit the requirements of conversational setting. The authors utilize the dialogues among the characters of the TV series 'FRIENDS' to create the relevant context and ask to complete the blank spaces with the name of the characters based on the context and utterances. In contrast to the above two datasets, it focuses more on the doer of the actions.

1.2 Challenges and Proposed Solution

This study aims to target and address the challenges posed by Conversational Machine Comprehension (CMC). CMC effectuate more challenges than simple machine comprehension task. The problems pertaining to CMC are discussed below:

i. Co-Reference Resolution Co-reference resolution is a classic problem in natural language processing (NLP) and is more difficult to address in CMC. Co-reference may not only appear in contextual situation but may also occur in question and answering as well. Co-reference can be categorized as explicit and implicit. In order to understand explicit co-reference, some personal pronouns known as explicit markers, are used. For example, in the given passage in Table 1.1 , the model has to identify 'his' in 'During his first season in Atlanta in 1998' corresponds to Galarraga. Similarly, to answer Q3 and Q4, the model has to refer to Q1 to understand that he refers to Galarraga. Implicit co-references are hard to decode without any explicit markers. Short questions with intention to seek more knowledge that implicitly based on previous context is referred as implicit co-reference. For instance, to figure out the complete meaning of Q3 *(When did he score the record?)*, the model should derive the co-relation between Q2 and Q3.

ii. Conversational History In machine comprehension task, questions and source passages are independent of each other and has no connection to the previously carried out question answering. However, conversational question answering does not work like this. The difference between machine comprehension and conversational machine comprehension is that questions in conversational machine comprehension form a series of conversation and the system requires a proper modeling of the conversation history in order to comprehend the context of current question correctly. The follow-up questions may be nearly related to the previous questions and answers. To understand it clearly, as depicted in Table 1.1, the machine needs to have knowledge of Q2 to answer the Q4 correctly. .

Out of these two challenges, we aim to target and address the problem of ***conversation history modeling*** into the system. CMC tasks, that combines conversational history into machine comprehension , aligns with the general process that how humans perceive the current question by

TABLE 1.1: Passage taken from QuAC dataset [11]

Passage	During his first season in Atlanta in 1998, Galarraga silenced his critics. He proved that he could still have great power production at lower altitudes, hitting .305 with 44 home runs and 121 RBI. This made Galarraga the first player in Major League history to hit 40 or more homers in consecutive seasons for two different teams. During 1999 spring training, Galarraga developed a sore back. Treatment from the team's trainers and team doctor included hydrobaths, massages, muscle relaxers, and stretching, but would not stop the nagging soreness. He was referred to a medical oncologist at Atlanta for a thorough physical exam and an MRI. When the diagnosis came in, the famous Galarraga smile disappeared. On his second lumbar vertebra in his lower back he had a tumor known as non-Hodgkin's lymphoma, a form of lymphatic cancer. He missed the entire 1999 season receiving cancer chemotherapy. Rockies third baseman and ex-teammate Vinny Castilla switched briefly from his traditional number 9 to number 14 on his jersey to honor Galarraga's cancer fight. Galarraga returned to the field in the spring of the year 2000 in high spirits and good shape after undergoing chemotherapy and a strict workout routine. In his third at-bat of opening day of the 2000 season, Galarraga knocked in the winning run with a home run, and he showed his big smile again. In April and May, he was tied for first place in home runs in the National League and he was batting .300. At the end of the season, Galarraga had batted .302 with 28 HRs and 100 RBIs. He was awarded his second National League Comeback Player of the Year Award by The Sporting News. Galarraga asked the Braves ownership for a two-year contract, but the most that it would offer was a one-year contract. Hence, Galarraga decided to become a free agent, and he signed with the Texas Rangers for two years.
Q1	Did Galarraga beat the cancer?
A1	Galarraga returned to the field in the spring of the year 2000 in high spirits and good shape after undergoing chemotherapy and a strict workout routine.
Q2	What records does he hold?
A2	He was tied for first place in home runs in the National League and he was batting .300.
Q3	When?
A3	April and May.
Q4	How many home runs did he hit?
A4	At the end of the season, Galarraga had batted .300 with 28 HRs and 100 RBIs.

considering the entire context. The ability to take into account previous utterances is key to building efficient machine comprehension systems that can keep conversational question answering active and useful [5]. Yet, modeling conversation history in an effective way is still an open challenge in such systems. Existing approaches have tried to address the problem of history conversation modeling by prepending history questions and answers to the current question and source passage [11]. Though this seems to be a simple method to improve the answer's accuracy, in reality it fails to do so. Other approaches used complicated attention mechanism [22] or complex Graph Neural Networks [6] to deal

with this issue. One recent work introduced the use of history answer embeddings [42] but they did not take the history questions into consideration to generate the context of the conversation. Also, these works lacks the approach of choosing the history that is relevant to the question and instead consider the entire conversation that may bring about the noise data.

To address this research gap, in this study, we propose an effective BERT-based framework which introduces the history questions to support and improve the effect of history answer embeddings when looking for an answer in the given paragraph. This approach suggest to prepend the history questions to the source paragraph along with the generated history answer embeddings to provide a complete input to our BERT-based model. By doing so, the suggested model would be able to generate better contextualized representation of the words, thus resulting in better answer span. Apart from this, we introduce a context-query relevance mechanism to calculate the similarity score between the two. This helps our model to only consider the history turns that are relevant to the current question.

The key advantages of our approach are that i) it is simple to implement practically and thus do not generate relatively high resource overhead, and ii) it provides the relevant picture of conversational history to our BERT-based model making the model capable enough to generate the contextualized embeddings more accurately. The implications of these facts can be shown from the results of extensive experiments that we carried out.

1.3 Thesis Organization

The following chapters provide details of the BERT-based framework in an attempt to capture and introduce the conversational history into the model. Before detailing the experiments, the Background chapter covers the general discussion about machine comprehension models and different techniques that are utilized to successfully perform the question answering task. The Literature Review chapter review the literature that specifically aligns with our field of study and discuss the pros and cons of previously introduced models in a chronological order. The Implementation Methodology chapter provides the details about the dataset used in the study, explains the architecture of our proposed model, how the input for the proposed model is tailored to suit the model's requirements, shed some light on implementation details, and discuss the evaluation metrics used to evaluate our model. The Evaluation Results chapter examines the performance of our model over the other state-of-the-art models, and also analyzes the different results obtained after conducting ablation studies, and the Conclusion and Future Work chapter summarizes the findings of this study, and examines future extensions that can be introduce to further improve the overall performance of our proposed model.

2

Background

2.1 Overview

This chapter provides an overview of the general research work carried out to keep the progress going on in the field of machine comprehension. The discussion starts with the general architecture employed usually in machine comprehension systems, followed by the techniques used in different modules of the system. The chapter discuss each module in detail and covers all the methods that could be used in each module. In the other end, the chapter covers additional methods that could be utilized to improve the overall performance of the system. The research works that are specifically similar to our approach are discussed in next chapter.

2.2 Generic Architecture

Figure 2.1 shows the general architecture of the machine comprehension systems. The system takes the provided context paragraph and query as an input and returns the answer as an output. It comprises of four prominent modules: Word Embeddings, Feature Mining, Context-Query Interaction, and Answer

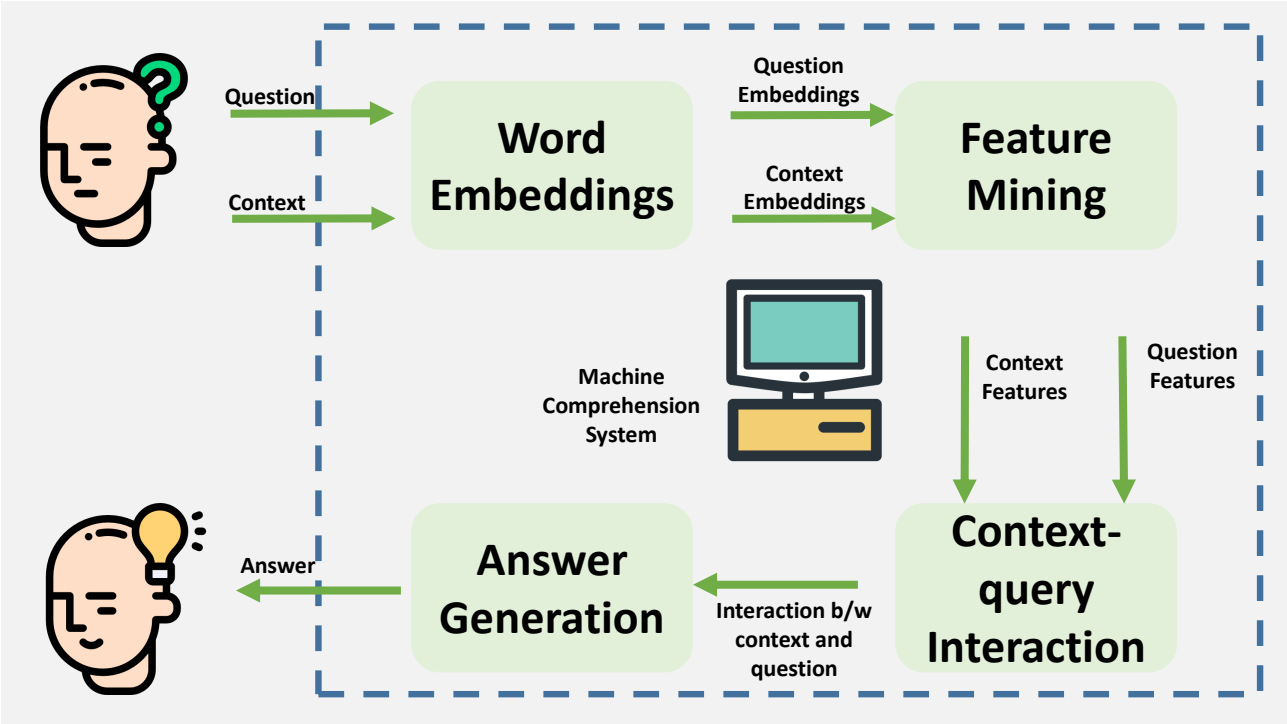


FIGURE 2.1: The generic architecture of machine comprehension systems.

Generation.

Unlike rule-based methods, deep neural networks and pre-trained language models show promising results in extracting contextual information which is very necessary for the good performance of machine comprehension models. In this section, we consider the different employed methods in the different modules of machine comprehension systems as shown in Figure 2.2. The key functions and the methods employed in each module are discussed below:

2.3 Word Embeddings

Since machine is incapable to perceive the human language, it is very essential to convert the natural language input into a form that is understandable by a machine. This is done by word embeddings module that converts the given text into a vector of fixed-length in the initial stage of the machine comprehension process. This module outputs query and context embeddings by taking query and context in textual form using various approaches. The traditional methods for word-representation such as Word2Vec [33] or one-hot embeddings are usually blended with other linguistic-based features, such as categorization of query, parts-of-speech or name-entity recognition, to extract syntactic and semantic information from the words. Moreover, the new trend is to use the contextualized word representations pre-trained on huge corpora to generate more meaningful results.

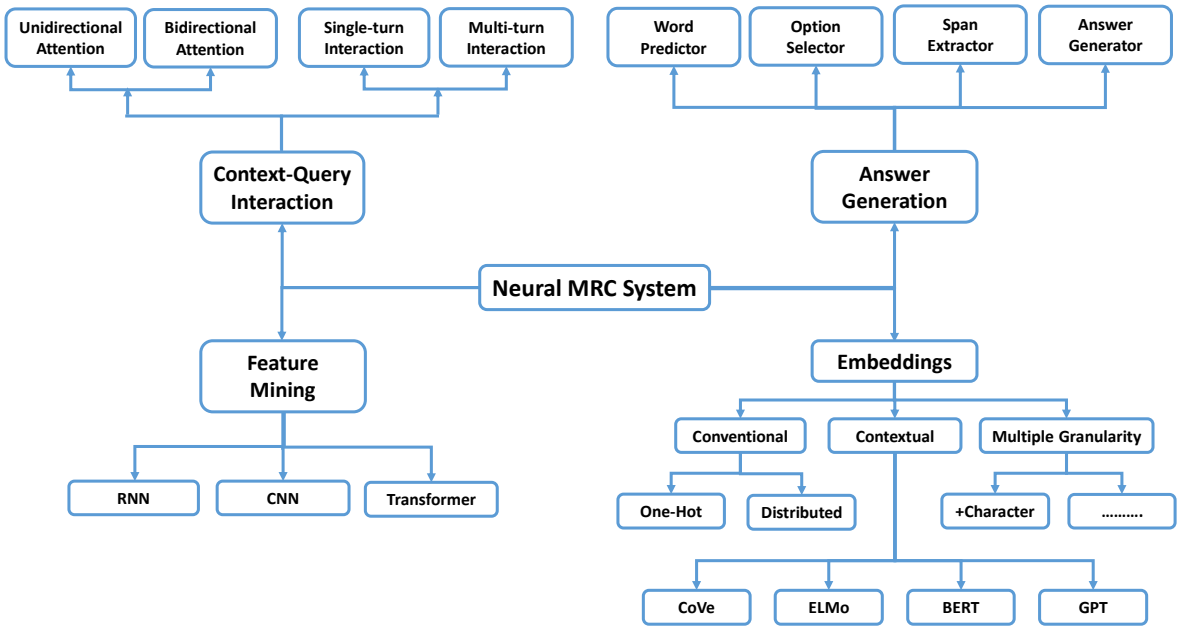


FIGURE 2.2: Typical techniques used in machine comprehension system [29].

2.3.1 Techniques for Word Embeddings Module

This module is responsible for encoding natural language input into fixed-length vectors that are understandable by the machine. As pointed out by Dhingra et al. [15], the choices made in word representations can make a noticeable difference in the overall output of the reader. Thus, the main task of this module is to determine that how to encode the given context and the query sufficiently. The approaches used to generate word representation can be categorized into traditional word representation and pre-trained contextualized word representations.

Traditional Word Representations

Traditional word representation includes one-hot embeddings which can be described as:

i. One-hot Embeddings This method uses binary vectors to encode the words which corresponds to the same size as that of the total words in the dictionary. Only one position is represented by 1, in these vectors, showing the correspondence to the word while the rest of the positions are filled up by 0s. It can be conveniently used in the early stage of the word representation as long as the vocabulary size is small. However, representing words using these embeddings is insufficient and may experience the problem of dimensionality when the size of vocabulary increases. Also, these embeddings are not capable of representing relationships among the words. For example, “bat” and “ball” both belong to the category of “sports” but it cannot be represented using one-hot embeddings.

Distributed Word Representations

To deal with the shortcomings of embeddings techniques like one-hot, the concept of distributed word embeddings is introduced. This method encodes the words into low-dimensional vectors in a way that closely related words are placed closer to each other in vector space so that co-relation between them can be identified easily. Several methods to generate distributed word representation have been proposed, out of which the popular and efficient are Glove [39] and Word2Vec [33] . In addition to machine comprehension, these vectors are also successfully applied in various NLP tasks such as sentiment analysis [1], machine translation [9], etc.

Pre-trained Contextualized Word Representations

Though the results achieved using distributed word representations were helpful in establishing a correlation between the words encoded in low-dimensional vectors, yet they cannot sufficiently extract contextual representation. To be concrete, for a single word the representation generated by distributed word representation is the same in varying contexts. To address this issue, the concept of contextualized embeddings is introduced by researchers. These embeddings are pre-trained on large corpora of text and are then utilized as either distributed word embeddings or fine-tuned according to the specific task needs. This comes under the category of transfer learning and has obtained astonishing results in various NLP based tasks. The most successful application of these embeddings has been in the field of machine comprehension. These pre-trained contextual embeddings can help even a simple neural network-based model achieve promising results. The success of these pre-trained embeddings has provided a sudden rise to the field of natural language understanding. These embeddings can be categorized as:

i. Context Vectors (CoVE) Inspired by the advantageous implementation in the field of computer vision, which shifts the knowledge gained by pre-training on large image corpus such as ImageNet to specific tasks, McCann et al. [32] propose to infuse the benefits of transfer learning into the downstream NLP tasks. The approach starts with using sequence-to-sequence models to train LSTM encoders on a large scale English-to-German translation dataset and utilize the results obtained by the encoder to the other downstream NLP tasks. As the task of machine translation restricts the words to be encoded in context, hence the final outcome of an encoder can be termed as context vectors (CoVE). To apply the concept of transfer learning in machine comprehension, the authors integrates the output of machine translation encoder with the pre-trained GloVE word embeddings to exhibit the presence of context paragraph and query, and provide them to the dynamic decoder and co-attention used in Dynamic

Co-attention Networks (DCN) [62]. The performance of DCN when combined with CoVE surpasses the actual model on the SQuAD [45] dataset, which shows the efficiency of pre-trained contextualized embeddings in NLP tasks. However, the pre-training task of CoVE asks for a huge amount of parallel corpus and its performance will not be up to the mark if the amount of data is inadequate.

ii. Embeddings from Language Model (ELMo) Proposed by Peters et al. [40], Embeddings from Language Models (ELMo) is a successor of CoVE. In order to get the embeddings of ELMo, they pre-trained a bi-directional language model (biLM). Unlike CoVE, these pre-trained embeddings overcomes the restriction of parallel corpus and can get more accurate representation of the words by combining the outcomes of all the layers of biLM into one vector and assign a weighting score that is task-specific rather than just using the results of the top most layer. Varying linguistic and syntactic information can be captured by different levels of LSTM states. When using ELMo in the field of machine comprehension, the authors selects an advanced version of BiDAF introduced by Clark & Gardner [12] as a baseline model and better the top-performing model by a good percentage on SQuAD dataset. These embeddings shows promising performance in many NLP tasks and can easily be integrated into the existing models. However, its capability is limited in a way by the inadequate feature extraction property of LSTM.

iii. Generative Pre-Training (GPT) A semi-supervised approach regarded as Generative Pre-training (GPT) [43] is a combination of unsupervised pre-training and supervised task-specific fine-tuning. Word representations pre-trained using this approach can be utilized in various downstream NLP tasks with just some little adjustment and modification. The main module on which GPT is based on is a multi-layer decoder of Transformer [56]. Mainly, multi-head self-attention is used by the decoder to train the model which allows it to capture longer semantic representation which is not possible in RNN based models. Depending on the task-specific needs, the pre-trained parameters are fine-tuned once the training phase is done. For machine comprehension tasks such as multiple-choice questions, the authors combine query and context along with the possible answers and process the sequence with the Transformer network. In the end, the model produces an output distribution over all the probable answers to identify the right answer. This architecture manages to improve the accuracy by 5.7% as compared to the top-performing models on the RACE [28] dataset. Observing the top performances demonstrated by pre-trained contextualized word representations, Radford et al. propose a successor of GPT, called GPT-2 [44], later. It is an improvement over the previous model and is pre-trained on a very huge corpus, WebText, with more than 1.5 billion parameters. In comparison to the previous model, the new model has an increase in layers of Transformer architecture from 12

to 48. Also, multi-task training is introduced in replacement of single-task training which makes the model more generative. Due to these reasons, GPT-2 achieves outstanding results even in the area of zero-shot learning. However, there is one limitation in both, GPT and GPT-2, models that they can capture the context only in one direction i.e. from left to right and not vice versa. This is a considerable shortcoming which can affect their performance in various downstream NLP tasks.

iv. Bidirectional Encoder Representations from Transformers (BERT) To address the issue of the unidirectionality of former architectures used in the pre-training of models such as GPT and GPT-2, Devlin et al. [14] introduce a novel architecture known as Bidirectional Encoder Representations from Transformers (BERT). This model introduces the tasks like next sentence prediction (NSP) and masked language modeling (MLM) during the training phase, which enables the BERT to encode the context from both the directions into the word representations. A general comparison of BERT with ELMo and GPT is shown visually in Figure 2.3. As Transformer architecture is not able to mine sequential information, Devlin et al. solved this issue by introducing position-based embeddings to capture the information about position of tokens. Due to the bidirectional property and powerful transformer architecture, BERT's performance exceeds the top-performing models in eleven NLP downstream tasks. The technical details of the model are given below:

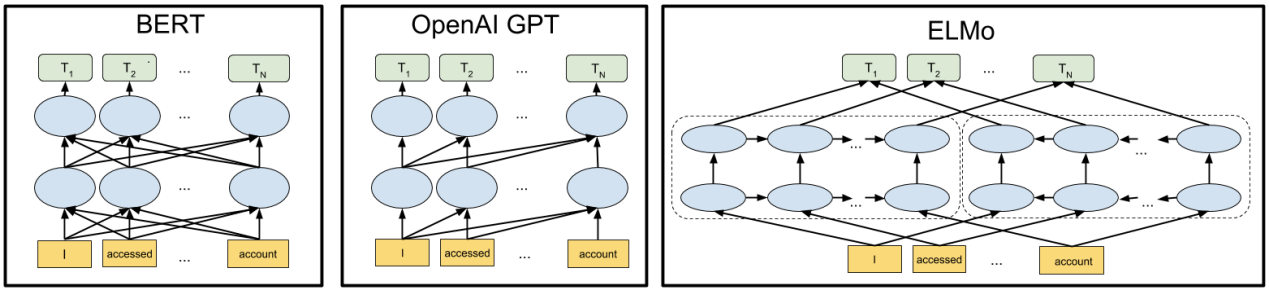


FIGURE 2.3: Comparison of BERT, OpenAI GPT and ELMo model architectures [14]

Input Representation: A BERT model can work effortlessly in both single sentence setting as well as a pair of sentences (one being a query and other being an answer) in one token sequence. At one instance for a given token, the input representation is a sum of corresponding token, position embeddings, and segment embeddings. The input representation can be visualized using Figure 2.4. The specifics are:

- The words are tokenized using WordPiece tokenization that has a token vocabulary of around 30,000. The split words are denoted by # #.

- The sequence length of upto 512 tokens is provided to support the learned positional embeddings.

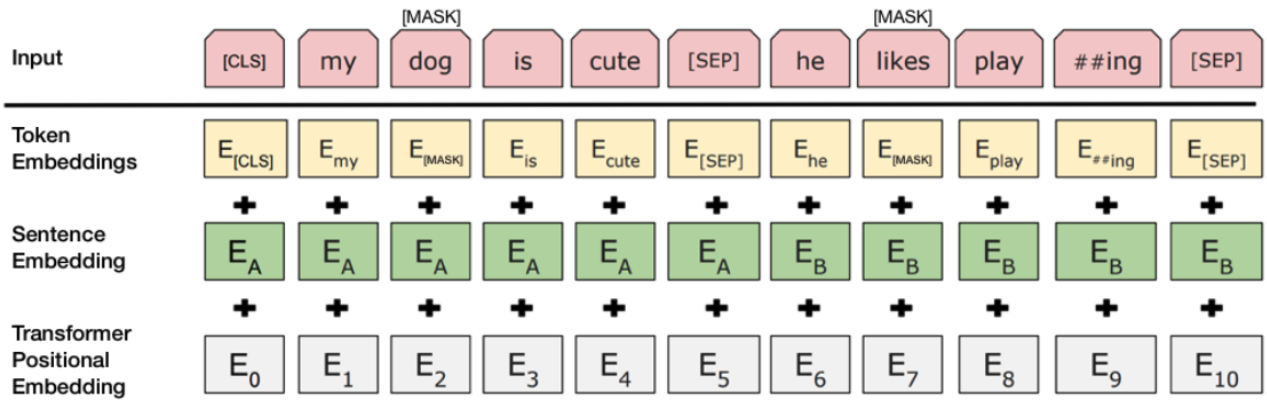


FIGURE 2.4: Input representation of BERT [14].

- Every sequence in a BERT model consists of a special token called special classification embedding [CLS]. The aggregate sequence representation obtained from the final hidden state of this special token is used in classification task for classification purpose. For non-classification tasks, this token is ignored.

- Sentence pairs are combine together in a single sequence and can be differentiated in two ways. First, they are separated using a special [SEP] token. Second, segment embeddings are used to categorize the sentences. Learned sentence A embedding is added to every token of the first sentence and a sentence B embedding is used for every token of the second sentence.

- For inputs based on single sentences, the model use only sentence A embeddings.

The similar architecture can be modified a little to adapt to any task-specific needs. For machine comprehension task, the architecture can be modified as shown in Figure 2.5. The final output of the model will be the answer spans of the given query.

Additional Granular Information

Pre-trained using Glove or Word2Vec, word-level embeddings are not capable of encoding rich linguistic and syntactic information, for example grammar, parts-of-speech, and affixes, which is very essential for deep learning models to clearly understand the context. To add the fine-grained semantic information to contextual word representations, several researchers propose methods to encode the

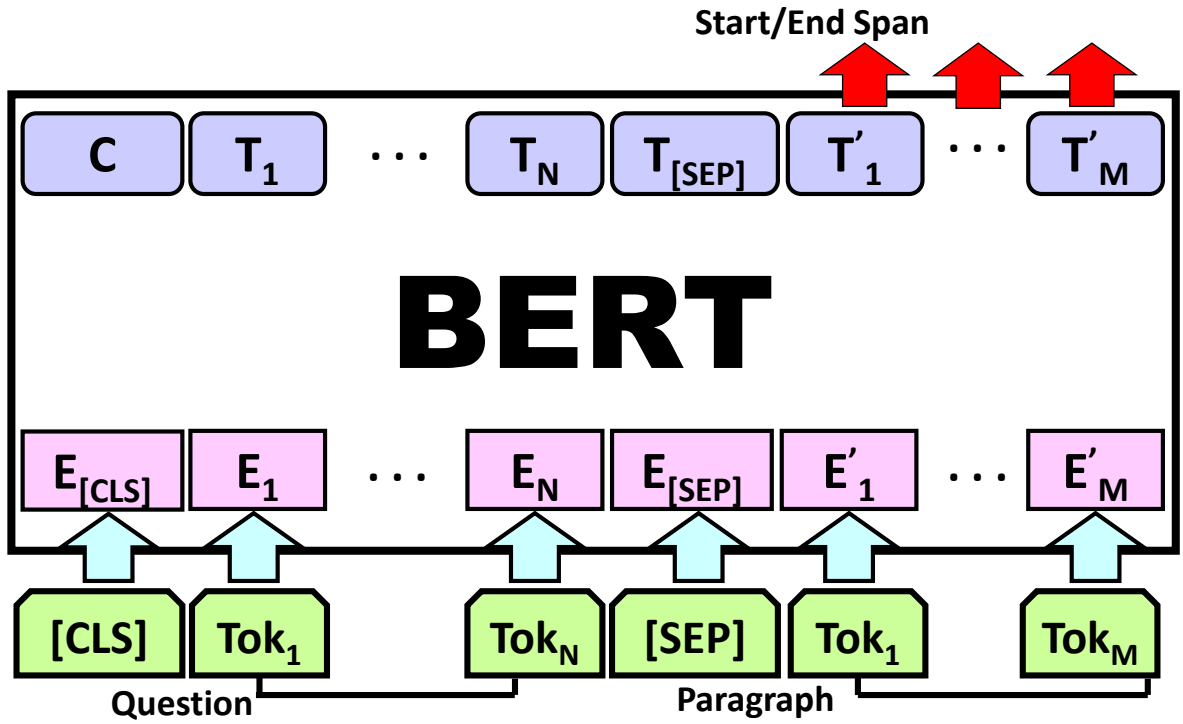


FIGURE 2.5: BERT adapted to suit Machine Comprehension task.

context passage and given query at different level of granularity. These levels of granularity are discussed below:

i. Binary Similarity Matching

This property is used to determine whether a word appearing in the given context is in the asked query or not and was initially employed in the traditional entity-centric model [4]. Later on, it was utilized in Word Embeddings module by several researchers to refine and enhance the representations of the words. The output of this property is 1 if the context word is an exact match of one word in the given query, otherwise the value of binary feature is considered to be 0. Also, the approach use partial matching to compute the relationship between the words of both context paragraph and query. For example, 'leadership' has partial similarity to 'leader'.

ii. POS Tags PoS or parts of speech is a common category of grammar to which a word is assigned in accordance with its syntactic function such as verb, noun, pronoun etc. Labeling words with their respective POS in NLP can represent complicated properties of the used word, which in turn, results in disambiguation. When converting into fixed-length vectors, parts of speech are treated as variables that are initialized randomly in the beginning and keep on updating when training.

iii. Character Embeddings The idea behind character embeddings is to represent a word on character-level. When put side by side to the word-level representations, they produce much better representation of sub-word morphologies and are able to mitigate the problem of out-of-vocabulary (OOV) words. Character-level embeddings were first used by Seo et al. [49] in the BiDAF model for the machine comprehension task. The model used convolution neural network to get the character-level embeddings. A fixed-length vector is used to encode each character in a word which is then provided as an input to CNNs as one dimensional vector. After applying the max-pooling function to the entire sequence, the results of CNN are character-level embeddings. The character-level embeddings and word-level embeddings are concatenated to be provided as an input to the next module. Apart from this, character level embeddings can also be encoded using LSTMs [21]. The outputs of previous hidden state are known as its character-level embeddings for each word. Furthermore, both, character-level and word-level embeddings, can be concatenated with a gated mechanism dynamically instead of using a plain concatenation method to eliminate the balance between infrequent and frequent words.

iv. Categorization of Query Categorization on the basis of query type (such as how, when, what, why, where) can give an additional piece of information that improves the process of answer prediction. For example, a query with 'who' provides more attention to the personal information. An approach was introduced by Zhang et al. [66] to represent various categories of query in an end-to-end training. The count of keyword frequency is obtained first in the model. Then, the information about category of query is encoded into one-hot vectors. The encoded information is then saved in a table. For every new query, they use the lookup method to access the table and utilize a feed forward neural network for the purpose of projection. The embeddings based on query categorization information are usually incorporated as an extra information into the word embeddings of given query.

v. Name-Entity Recognition The concept of name-entity recognition was first introduced in the community of information extraction and refers to the real world objects, such as people, place, organizations etc., with a particular name. Name entities are used as candidate answers when enquiring about these objects. Thus, incorporating the information about name-entity candidates can results in better chances of correct answer prediction. The approach use for encoding the name-entity tags is same as the POS encoding as mentioned above.

Word embeddings discussed in the above section can be integrated freely in the first module of any machine comprehension system. Hu et al. [21] utilize the character-level embeddings, word-level embeddings, name-entity tags, query-category embeddings, POS tags, and binary similarity matching in their Reinforced Mnemonic Reader [21] to introduce linguistic and syntactic information to

the contextual word representations. Extensive experimental results shows that adding extra information to word representations results in a deeper and better understanding of the context by any machine comprehension model that eventually leads to better prediction of an answer.

To conclude, the word embeddings generated using distributed word representation are the core of embeddings module. Since introducing additional representations with linguistic and syntactic information plays a significant role in improving the performance, the use of granularity representations have become very popular. As far as contextual word representations are concerned, they can drastically improve the model's performance, which can be utilized with other representations either solely or in combination.

2.4 Feature Mining

The output of the word embeddings module is then provided as an input to the feature mining module. To create a suitable understanding of the given passage and the current query, the goal of this module is to mine more hidden contextual information.

2.4.1 Techniques for Feature Mining Module

The output of this module is utilized by the next module i.e. context-query interaction as an input. The strengths of architectures like Convolution neural network, recurrent neural network and Transformer model are leveraged in this module, the description of which is provided in this section.

Convolution Neural Network (CNN)

The application of convolution neural network is very common in computer vision. When applied in the field of natural language processing, one dimensional CNN show astonishing results in extracting local contextual information with sliding window. Each layer in convolution neural network uses different level of feature maps to mine local features in varying window size. Then, the outputs of these layers are provided as an input to the pooling layers to lower the dimensionality but it retains the information, that is significant, to the maximum possible level. Using average and maximum operations to the outputs of each filter are popular ways to perform the pooling function. Although both n-grams and CNN can concentrate on the local features of a given sentence, the size of training parameters in n-gram models increases exponentially as the size of the vocabulary grow larger. On the other hand, CNNs are able to extract local information efficiently no matter what the vocabulary size is, for there is the representation of every n-gram in CNN is not necessary. Another great property

of CNNs is that they can be trained parallel which makes the processing time faster. The one main weakness of CNN is that they can just mine the local information and are not able to handle the long sequences.

Recurrent Neural Network (RNN)

RNNs have been known for showing astonishing results with sequential information for a long time. RNNs are regarded as recurrent networks as the outputs of the current layer are dependent on the outputs from the previous layers. The strengths of RNNs have been commonly applied to the various NLP tasks like sequence tagging, machine translation, and question answering. Especially the variants of RNN such as Gated Recurrent Units (GRUs) [9] and Long Short Term Memory (LSTM) [20] are much better at understanding long-term dependencies as compared to the vanilla ones. These variants are good at addressing the problems of gradient explosion and vanishing. Since encoding the previous and succeeding words are equally important to capture the accurate context, many researchers use the bidirectional RNNs to generate the embeddings of both context and query in machine comprehension systems. The feature mining process using bidirectional RNN could be either sentence-level or word-level.

As the sequence of sentences in machine comprehension is generally very long, several researchers prefer using the method of word-level embeddings of feature extraction to represent sequential information. Although, RNNs are good at capturing long-term sequences but the process of training is quite time-taking and exhaustive as the parallel processing of the sequences is not possible.

Transformer

Introduced by Vaswani et al. [56], Transformer is a deep neural network based model that has demonstrated outstanding performance in various NLP downstream tasks. Unlike RNNs or CNNs, which are based on recurrence and convolution respectively, the main idea behind Transformer model is the use of attention mechanism. It not only outperforms in alignment but also has the ability to parallelize owning to its multi-head self-attention mechanism. When compared to CNN, it provides more attention to the globalized entities, while requires less time to train in comparison of RNN. However, it is not able to keep the record of the order of the sequence without convolution or recurrence. To introduce the positional information, the authors introduce the addition of positional embeddings calculated by cosine and sine functions. The concatenation of word embeddings and position embeddings are provided as an input to the Transformer model. Practically, model usually stack several blocks with feed-forward network and multi-head self-attention .

QANet [64], proposed by Yu et al., is a classical representation of Transformer based machine comprehension system. The architecture of encoder block of QANet is very simple and innovative that integrates multi-head self-attention introduced in the Transformer architecture with convolutions. The experimentation outputs reveals that the QANet architecture on SQuAD obtains same accuracy as popular recurrent models with much improved inference and faster computational time.

Generally, most of the machine comprehension systems use recurrent neural networks in this component mainly due to their capability of handling longer sequences very well. Moreover, to speed up the training time, several researchers replace RNNs with either CNNs or Transformer. The advantage of using CNN is that they are parallelized and can get better local information with feature maps in varying sizes. On the other hand, the Transformer can eliminate the side-effects of both long-dependency and enhance the computational efficiency.

2.5 Context-Query Interaction

The relationship between the given context paragraph and the current query plays an important part in finding the right answer. Given this information, the system is efficient enough to analyze which sections of the context are more significant to the query and could result in an accurate answer.

2.5.1 Techniques for Context-Query Interaction Module

Influenced by Hu et al. [21], current researches can be categorized into two classes depending on system's approach to mine the relationship between the context and query; single-turn interaction, and multi-turn interaction. Despite the type of interaction method model use to identify the correlation, attention mechanism play a significant role in asserting which sentences of the given context paragraph are highly relevant to the query.

Imitated from human's instinct, attention mechanism was first applied in the field of machine translation and showed a remarkable achievement on dynamic token alignment [30]. Later on, known as an effective and simple method to encode sequential data along with its significance, attention gained massive popularity and significant improvement in natural language processing tasks such as sentiment analysis, text summarization, semantic parsing, etc. This module plays a significant role in correctly identifying the relationship between the context and query. In the field of machine comprehension, the concept of attention mechanism can either be unidirectional or bi-directional depending on how it is utilized. In the following section, we first shed some light on the two different

categories of interaction, followed by the discussion on unidirectional attention and bidirectional attention.

Single-Turn Interaction

This basic architecture lets the computation of interaction between the context passage and query only once. In the early days of machine comprehension, the interaction between context and query was based on such single-turn architectures in many machine comprehension systems such as the Attention Sum Reader [26], the Attention over Attention reader [13], and so on. Although such interaction can perform reasonably good in handling small cloze test tasks but when the query needs referring to a number of sentences in the context for the reasoning process, it is difficult for this approach to identify the right answer.

Multi-turn Interaction

Unlike single-turn interaction, the architecture of multi-turn is complex and tries to imitate the process of human re-reading provided the memory of context paragraph and query. When talking about multi-turn interaction, the fact that whether the system can keep track of the previous state or not has direct effects on the performance of the next iteration. Multi-turn interaction can be carried out in three ways:

One of these methods computes the similarity value between the context paragraph and query based on the past attentive representations of context. In the Impatient Reader Model [19], the context representations keep on updating dynamically by this method as each token based on query comes in and read by the model. This encourages the phenomenon of human re-reading of the given context with the presented query.

Second in line is the approach that suggests the use of external memory slots to save previous memories. The method is utilized by the Memory Networks Model [60] and it can explicitly store long-term memories along with the easy access to reading memories. The machine comprehension systems can deeply understand the relationship between the context paragraph and query by multiple turns of interaction using the mechanism introduced by Weston et al. [60]. After having the context information as an input, the model stores the information into the memory slots and updates them automatically from time to time. The process of finding the answer to the query involves identifying the section that is most relevant to the query and converts it into the required answer representations. Though the concept perform well in overcoming the issue of insufficient memory but is difficult to be trained via back-propagation. An end-to-end version of memory networks [51] is introduced to

address this shortcoming. Explicit memory storage is embedded with continuous representation as an improvement to previous models. Furthermore, the process of reading and updating the memory information is shaped by the neural networks. This development in memory networks significantly reduce supervision during training and can be applied to various tasks. The property of memory networks of updating the memory in multiple-hops makes it popular in machine comprehension systems. Another model referred to as MEMEN [38] is proposed that stores both context-aware representations of query and query-aware representations of context along with the representations of the candidate answers into memory slots and updates them dynamically from time to time. Likewise, another approach [10] utilizes external memory slots to save query-aware representations of the context and update memory slots with bidirectional GRUs.

The third method leverages the benefits of the recurrence feature of RNNs, utilizing the hidden state to save all the interaction information of the past. Match-LSTM, propose by Wang & Jiang [58], is used recurrently to perform multiple interactions. This model, originally proposed for text entailment, when applied in the field of machine comprehension can simulate the process of reading passages with query information. First of all, the approach uses the common attention mechanism to compute the attentive weights of each token of the context to the given query. After computing the dot product of attentive weights and query tokens, the model combines it with the context tokens and pass it on to the match-LSTM to obtain query-aware representations of the context paragraph. Likewise, the process is carried out in the reverse direction as well to get fully encoded contextual representations. In the final step, the results of match-LSTM from both the directions are joined together and are then used for predicting the answer by answer prediction module. Models like IA reader [50], R-net [59] and Smarnet [7] also use RNNs to update the query-aware context representations to perform multi-turn interactions.

Some previous works considers both the context words and query words equivalently when extracting their inter-relation. But ideally, the most significant part should be given more attention for improved context-query interaction. Gate mechanism, which controls the amount of mutual information between context paragraph and query, is a significant component in multi-turn interaction.

The Gated-Attention (GA) reader [16], apply the gate mechanism to determine how the information stored in the given query has an affect on the context words when updating the context representations. This method is applied using element-wise multiplication between query embeddings and the intermediary representations of the context words multiple times.

Unlike GA reader, the Iterative Alternating Attention Mechanism [50] updates the representations of both context and query. Query representations are updated in accordance with the

previous search state. As far as context representations are concerned, these are not only updated with respect to the previous reasoning process but also in accordance with the current query. Then, the gate mechanism, which is implemented using a feed-forward network, is employed to calculate the similarity score between the two. This method is capable of mining the relationship between the context paragraph and query alternately.

The Smarnet [7] model not only utilize the gate mechanism to control the effect of query on the given context but also propose the use of additional gate mechanism to refine the query representations with the information provided in context paragraph. Combining the two gated mechanisms enables the reading of query and context paragraph on an alternate basis with common information.

Earlier architectures ignored the fact that context words have varying significance to answer different queries. To resolve this issue, Wang et al. [59], in their R-net model, propose the gate mechanism to eliminate the irrelevant parts from the context and focus on the significant parts to answer the query. This approach can be viewed as an extension of the attention-based recurrent network. In contrast to the match-LSTM, this approach proposes the use of an additional gate mechanism based on the current context representations. Also, as RNNs based models cannot work well with the lengthy documents due to the lack of sufficient memory, this model adds self-attention to the context itself. This approach filters the context representations dynamically based on the common information from both the context paragraph and query.

To conclude, single-turn interaction may fail to incorporate the complete information that represents the relationship between the context and query. This weakness has been overcome by multi-turn interaction that has the memory of previous context and query which enables it to deeply extract the mutual relationship between the two and combine evidence for answer prediction. Thus, the methods that are used in capturing the context-query interaction have a very significant effect in answering the query.

Unidirectional Attention

The flow in unidirectional attention usually starts from query and ends at the context, highlights the most significant sections of the context passage according to the given query. It is normally believed that if the words in the context are more relevant to the query, they are most likely to be the part of the answer. The similarity of context embedding C_i and the whole query representation Q is computed using $S_i = f(P_i, Q)$, where f is a function to calculate the similarity between the context and query. The softmax function is used for normalization and then attention weight α_i for each context word is

obtained, which is further utilized by the machine comprehension system to predict the answer.

Thus, the unidirectional attention mechanism can easily focus on the significant parts of the context that plays an important role in answering the query. However, this method does not take into account the words belonging to the current query which are also very crucial for finding the answer. Hence, this method is not sufficient enough to identify the mutual relationship between the context paragraph and query.

Bidirectional Attention

To address the above-discussed weaknesses, the concept of bidirectional attention flow is introduced, that not only calculates the relevance from left to right direction but also vice versa. This mechanism, that considers the input from both forward and backward direction, provides a piece of additional information to process the context and query.

To get the pairwise matching matrix *M(i,j)*, the similarity score between the context embeddings C_i and query embeddings Q_j is calculated first. The result of column-wise softmax function is then considered as query-to-context attention weight α, whereas β is regarded as the context-to-query weight calculated using row-wise softmax function. In the AoA reader (short for attention over attention reader [13]) architecture, the dot product between context embeddings and query embeddings is computed to get the similarity matrix *M(i,j)*. Then, the context-to-query based attention and query-to-context based attention are computed. To combine these attentions, Cui et al. [13] propose the concept of attended attention which is computed using the dot product of α and average result of β, which is later used in predicting the answer.

In order to attend to the paragraph and query at the same time, Xiong et al. [62] combine the two directional attention as follows:

$$C = \alpha[Q, \beta P], \tag{2.1}$$

where *C* denotes the co-attention representations which are based on attention information of both context paragraph and query. Later Xiong et al. [61] propose DCN+, an enhancement of DCN, introducing a residual connection to combine the results of co-attention to encode richer information to the sequences of input. Rather than directly using the attention weights for the answer prediction like AoA reader, this approach further calculates context representations using two directional attentive information.

Unlike DCN and AoA, which summarizes the outputs of bi-directional attention flow directly, Seo et al. [49] allow attentive vector flow into another layer of RNN to generate query aware

representations of the context paragraph, which in turn can lessen the loss of information caused by summarization that took place early.

To conclude, machine comprehension systems usually employ unidirectional attention mechanism i.e. query-to-context attention at the early stage to identify which sentences of the context paragraph are more significant to answer the current query. However, using only query-to-context attention is not an optimal solution to identify the correlation between the context and query. To overcome the shortcoming of uni-directional attention, the concept of bi-directional attention is introduced and widely applied, which also considers the aspect of context-to-query attention and generate attentive representations with the combination of the context and query information.

2.6 Answer Generation

This is the final component of the machine comprehension system, which results in an answer to the query provided the complete information accumulated from the previous modules. However, the output of this module is not uniform as it is further sub-divided into different categories. The type of answer prediction varies with the machine comprehension task. The tasks can be defined as:

i. Cloze Test The cloze test is inspired by an exam scenario where students are tested for their language proficiency and is sometimes known as the gap-filling test. The task is used to measure the machine's capability to understand the natural language. In this task, some words or entities are removed from the given passages. The aim is to identify the missing ones and fill in the gaps. There is a possibility that some tasks may provide options but again this is not true in every case. Cloze tests require a clear perception of the context and the utilization of vocabulary, and can be asserted as challenging for machine comprehension. It can also be regarded as a word or entity prediction task owning to the fact that the answers in this task are either words or entities.

ii. Multiple Choice Questions The task of multiple choice question is an another example of machine comprehension that takes its inspiration from a language proficiency exam setup. The objective is to identify the right answer to the query from the candidate answers given the context paragraph. It is mandatory for this task to have options as possible answers to have the right selection. Unlike the cloze test, answer options for this tasks are not restricted to words or entities and are more free-form.

iii. Span Extraction Although cloze test and multiple choice questions are good for measuring the machine's capability of natural language understanding, yet there are certain limitations pertaining to these tasks. To state more clearly, entities or words are insufficient to provide answers to the questions. At times, complete sentences are required. Also, candidate answers are not provided in some scenarios. This weakness can be addressed by the span extraction task. Provided the context paragraph and a query, the goal of this task is to extract the span of a text containing the answer from the given context.

iv. Free-Form Answering Span extraction task made great progress in advancing the machine's ability to give more flexible answers as compared to cloze test and multiple choice. But still, it is not realistic to restrict the answers to a limited span of text in a given context. To predict a more accurate answer, the machine needs to reason across the multiple contexts and epitomize the findings. This can be achieved by free answering. Out of these four tasks, the most challenging is free answering as there is no restriction on the form of an answer and is more relevant to the real-world applications.

2.6.1 Techniques for Answer Generation Module

As discussed earlier in Section 2.2, machine comprehension tasks are categorized into free answering, cloze test, span extraction, and multiple-choice, the methods for answer prediction could be answer generator, word predictor, span extractor, and option selector respectively. These methods are briefly discussed as:

Word Predictor

The objective of the cloze test task is to find the missing entity or word from the given paragraph and fill in the blank to complete the sentence as the answer. In early works like an Attentive Reader [3] directly uses the query-aware representations of the context to find the answer, which improves the performance significantly and makes the prediction process easy.

The methods discussed above utilize attentive representations of the context to predict the correct one word answer but they cannot ensure that the selected word is chosen from the context paragraph, which does not satisfy the criteria of cloze tests. In order to overcome the issue that selected answer word may not be in the given context, Attention Sum (AS) reader [26] based on the concept of pointer networks is proposed. Pointer networks, put forward by Vinyls et al. [57], is designed for the tasks where the output is selected from the provided inputs and can well-satisfy the requirements of the cloze test. The AS reader does not calculate the attentive representations, rather it directly uses attention weights to select the right word form the context. The outputs generated using attention

mechanism for the same word are combined together and the right answer is choosen on the basis of highest value. This approach is simple as well as adequate for cloze test tasks.

Option Selector

The goal of option selector is to select the right answer that fits the situation from the given answer options. The most popular way is to compute the relevancy score between the candidate answer representations and attentive context representations and the option with the higher similarity value is selected as the correct answer to the query.

Chaturvedi et al. [2] use convolution neural networks to encode the information about the query, given candidate answers and relevant sentences from the context. Then, the cosine similarity function is used to compute the similarity between them. The answer with the most relevance is chosen as the final answer. Later, Zhu et al. [69] propose the use of information about answer options to mine the correlation between query and the context. The bilinear function is used to assign a score to each option according to the attentive information in answer prediction module. Finally, the answer with the highest assigned value is predicted as the right answer. The convolution spatial attention model [8] calculates the similarity value among query-aware candidate representations, self-attended query representations and context-aware representations with dot product to completely understand the relation between the given query, candidate answers, and the context. The varied similarity scores are combined and then pass on as an input to the convolution neural networks with different kernel sizes. The outputs produced by convolution neural networks are known as feature vectors and are fed to the fully-connected layer to compute a score for each option. The option with the highest value will be selected as the right answer.

Span Extractor

The aim of the span extraction task is to retrieve a chunk of sequence from the given passage instead of predicting a one word answer. This task can be termed as an improved version of the cloze test task which requires the prediction of one word as an answer. The word predictor methods that were once used for cloze test tasks do not apply to span extraction as they only predict a single word whereas span extraction requires the prediction of a sequence of words. Inspired by the idea of Pointer Networks [57], Wang & Jiang [58] put forward two different models, the Boundary model, and the Sequence model, to address the weaknesses of word prediction approaches. The outputs of the sequence model are the positions where the answer tokens appear in the context. The answers generated by this model are not necessarily a consecutive span and might not be a subsequence of the given context paragraph.

This drawback is improved in the Boundary model, which only results in the start span and end span of the answer. The boundary model is simple to implement and demonstrates superior performance on the SQuAD dataset [45].

Keeping the possibility in mind that there could be more than one probable answer text in the given context, and the boundary model would retrieve the wrong one with local maxima. To address this issue, Xioang et al. [62] put forward a dynamic pointing decoder to choose the starting and ending span of an answer using multiple iterations. This approach utilizes LSTM to predict the two spans based on representations corresponding to previous answer prediction. To predict the answer spans in the given context, the authors introduce Highway Maxout Networks (HMN) with Highway Networks [17] and Maxout Networks [18] for varying context topics and query types.

Free-Form Answering

Owing to the advancements in the technology and machine comprehension tasks, the answers to be predicted are no longer restricted to be a part of the given context, rather they need to be generated from both query and context. To be concrete, answer expressions may differ from the evidence passage of the context or may be based on multiple context paragraphs. Answers generated using free-form answering have the least possible restrictions, which in turn, pose strict requirements for the answer prediction module. In order to cope with this challenge, several answer generation techniques are proposed to generate free-form answers.

One approach, S-Net [53], performs the extraction and synthesis process in answer generation module for free-form answer generation task. The extraction module used is derived from the R-net [59] whereas the generation module utilizes sequence-to-sequence architecture. In short, bidirectional GRU is used as an encoder to generate the representations for both context paragraph and query. Particularly, start and end span predicted from the snippet of the given context are incorporated with context representations as additional features. As for decoder, GRU's state is updated based on the context representations generated previously and attentive intermediate representations. In the end, softmax function is applied and the output of the decoder is regarded as the final answer.

The development of the answer generation module has successfully overcome the weaknesses of the extraction task and generates more flexible and free-form answers. However, these answers may suffer from the problems of illogical problems and syntax errors. Thus, generation and extraction modules are put to work together to provide complementary information for each other. For instance, the extraction module of S-net first labels the approximate boundary of the answer span while the generation module is responsible for generating answers not restricted to the context of

the evidence paragraph. Though extraction approaches have performed reasonably well in machine comprehension systems, the answer generation approaches are not very popular yet.

2.7 Other Popular Techniques

Several techniques are used along with the above discussed methods to increase the chances of answer's accuracy. Some of which are:

2.7.1 Answer Ranker

In order to validate whether the system has predicted the correct answer or not, several researchers propose the use of a module named answer ranker. The mechanism behind this module is that some options as probable answers are first retrieved and the answer with the highest score is regarded as the correct answer. Another model called EpiReader [54] joins the pointer method with the answer ranker module. It first similarly mines the answer candidates as AS Reader [26], extracting the answer span with the highest value of attention sum score. Then, EpiReader pass these options to the reasoning module, which uses the hit and trial system by putting the options to the query sequence at placeholder positions and calculates their score in terms of probability to be the correct answer. The option that gets the score greater than the rest of the options is selected as the right answer.

To retrieve the options with different lengths, Yu et al. [65] suggests two methods. In the first approach, they apprehend the parts of speech (POS) patterns in the training set and choose subsequences in the evidence passage that matches the pattern as the given options. The second approach computes all the possible answer spans of a fixed length from the given context. After getting all the options, the method calculates the similarity score of obtained spans with query representations and selects the highly relevant one as the answer. The accuracy of answer prediction has enhanced with the ranking mechanism. This method has also encouraged the several researchers to detect unanswerable questions later and much progress is made in the area of answering unanswerable queries.

2.7.2 Reinforcement Learning

Most machine comprehension systems just use the estimation of maximum likelihood when training the model. However, there exists no connection between evaluation metrics and optimization objective. This results in candidate answers, which precisely matches the ground-truth or are somehow similar to the ground truth but are not located at the labeled position, being ignored. Moreover, if the span

of the answer is too long or has an unclear boundary, models would also fail to identify the right answer. For machine comprehension models, evaluation metrics such as F_1 or exact match (EM) are not differentiable, therefore, several researchers propose the idea of introducing reinforcement learning to the training process. Hu et al. [21] and Xiong et al. [61] both use F_1 as a reward function and tackle both reinforcement learning and maximum likelihood estimation as a multi-task learning problem. The proposed approach can take both textual similarity and positional information into consideration.

2.7.3 Sentence Selector

Practically, if a machine comprehension system is provided with a lengthy context paragraph, it is hard and time-taking for it to comprehend the complete paragraph to answer the query. However, identifying the sentences that are more suited and relevant to the query beforehand is a probable way to speed up the process of training. With this concept in mind, Min et al. [34] introduce the process of sentence selection to extract a significant set of sentences that contributes more to finding the right answer to the given query. The architecture used for the implementation of sentence selector is sequence-to-sequence, which has an encoder to generate encodings for both sentences and query, and a decoder that calculates the score for each sentence by computing the similarity between the sentence and query. If the value is greater than the pre-set threshold value, the sentence is selected to pass on to the machine comprehension system. Using this method, the number of selected sentences varies depending on the given query.

Machine comprehension systems with sentence selection module provides the reduction in the inference and training time with either equivalent or better performance than the systems without the sentence selector module.

2.8 Conclusion

In this chapter we discussed the four modules that generally constitutes a machine comprehension system: Word Embeddings, Feature Mining, Context-Query Interaction, and Answer Generation. We also illustrated and discussed the methods and techniques that are usually employed in these modules to successfully carry out the task of machine comprehension. The next chapter highlights the research works that very specifically aligns with our research direction.

3

Literature Review

3.1 Overview

A typical machine comprehension model consists of 4 components that are Word Embeddings, Feature Mining, Context-Query Interaction, and Answer Generation. As we have discussed in detail in Chapter 2 that for each of these modules, different methods and techniques are available. Different models employ different set of techniques according to the needs of the system. For our model we have used pre-trained contextualized word representations generated using BERT for Word Embeddings module which is based on Transformer's encoder [56] with bidirectional attention to handle context-query interaction mechanism and uses span extractor [45, 25, 54] method in answer generation module. As far as dataset is concerned, we have used Question Answering in Context (QuAC [1]). The reasons for using QuAC dataset and its comparison with other datasets are provided in detail in Chapter 4. In this chapter, we discuss the research works that aligns with our research direction and identify the gaps in the presented literature. In the next chapter, we try to address the identified loopholes with our model. For better understandability, we divide the works on the basis of techniques used:

[1] http://quac.ai/

3.2 Bidirectional Attention Flow for CQA

The concept of bidirectional attention flow was introduced in 2017 by Seo et al. [49] in the form of BiDAF network for the task of machine comprehension. It achieved state-of-the-art results on SQuAD [45] and CNN/Daily mail [35] cloze test and is considered as the baseline model for machine comprehension tasks. It is based on a hierarchical multi-stage architecture as shown in Figure 3.1 for modeling the word representation of the given context passage at different levels of granularity such as word level and character level. The model uses bidirectional attention flow to capture the query-aware context representations. The model provides improvement in attention mechanism over previously popular attention paradigms. The attention mechanism utilized in the model is not used to summarize the context of the given passage into a fixed-length vector. Instead, the attention is computed at each time step. The attended vector at each time step along with the word representations from the previous layers is allowed to flow through the successive modeling layer. This cut down the risk of information loss due to early summarization process. Secondly, they utilize a memory-less attention mechanism that is the attention at each time stamp is a function of current question and context paragraph and does not directly relate with the attention during the previous timestamps. This method allows the attention of current time stamp to remain unaffected by the inaccurate congregation at previous timestamps. Third, the model employs the concept of utilizing attention mechanism in both directions, context-to-query- and query-to-context, which provides additional information to each other. This model introduce the bidirectional attention flow to the task of machine comprehension and is further improved by different authors to predict the answers more accurately.

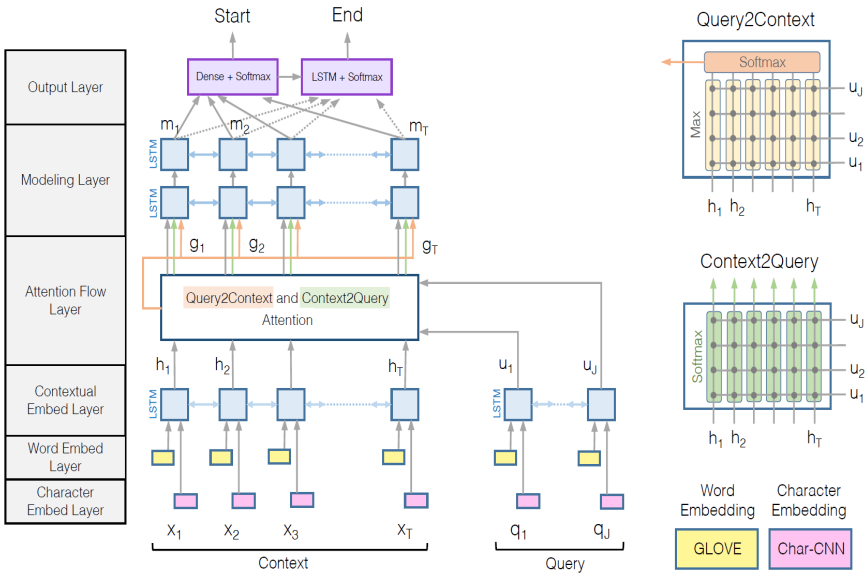


FIGURE 3.1: The working architecture of BiDAF [49]

BiDAF++ [12] is an improved version of BiDAF model that adds a self-attention layer after the bidirectional component, simplify some of the pooling operations and replace the LSTMs with the gated recurrent units (GRUs) as shown in Figure 3.2. In addition, this model introduce an innovative type of deep contextualized word embeddings that represents both i) complex characteristics of word use, and ii) how the use of these differ across the different contexts. These embeddings are called Embedidngs from Language Models (ELMo) as they are the learned function of the internal states of a bidirectional Language model (biLM), which is pre-trained on a huge text corpus. Introducing these embeddings to the baseline model, the F_1 score improved by 4.7% which results in relative error reduction of around 24.7%.

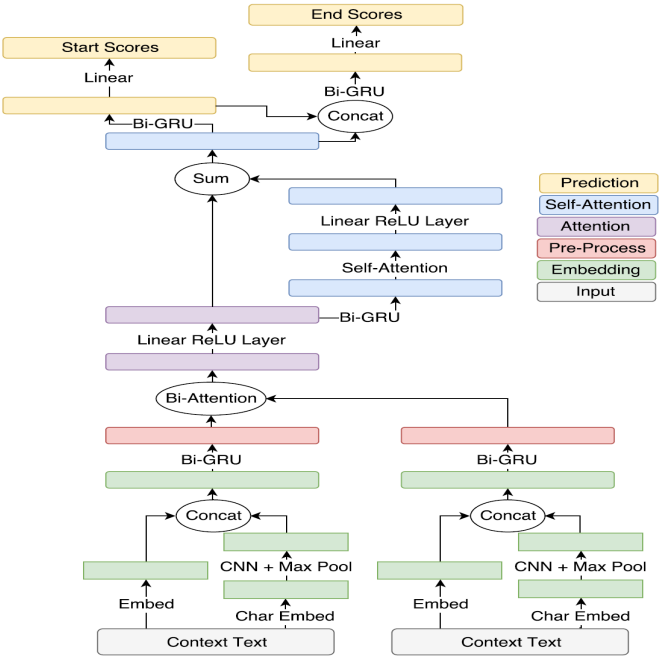


FIGURE 3.2: BiDAF++ model [12]

Although BiDAF++ performed very well with single-turn machine comprehension and achieved state-of-the-art results on SQuAD dataset. But when the same model was applied to conversational machine comprehension, BiDAF++ failed to model the dialog context. So, BiDAF++ w/k-ctx [11] is introduced to overcome the modeling of history modeling. The model modify the process of generating embeddings for query and passage to integrate the dialog history. They not only addressed the problem but also released a conversational machine comprehension dataset known as Conversation Answering in Context (QuAC) [11] for the public. The role of context in conversational question answering is very significant for understanding the current question. The prediction of answer can be influenced by the previously asked questions.

The model is tested by incorporating different number of history turns into the input and

the experiment results show that integrating previous conversation significantly improve the result and is essential for solving the task.

3.2.1 Weaknesses

- Though BiDAF and BiDAF++ perform very well but these models do not take previous context of the conversation into consideration which made it difficult for the system to decipher the current query.
- BiDAF++ w/k-ctx incorporate the context until 3 turns instead of sudying the effects on including the entire conversation. Also, they only utilize answer embeddings from the given context rather than incorporating the entire history turns.
- Also, BiDAF++ w/k-ctx utilize ELMo in their model to generate the word representations. These embeddings are not deeply bidirectional, hence are not capable to capture the true essence of the words.

3.3 BERT-Based CQA

Devlin et al., in 2018, put forward a bidirectional language model called Bidirectional Encoder Representations from Transformer [14]. The model utilizes Transformer's encoder for language understanding and has achieved state-of-the-art results on 11 NLP downstream tasks. Chen Qu et al. for the first time try to implement the CMC task with BERT as a base architecture. Previous model use a very simple approach for adding the dialog history i.e they prepend the history conversation as a part of input. This model propose a new way of introducing history turns to the model by giving tokens extra embedding information. They introduced an additional History Answer Embeddings (HAE) layer into the traditional architecture of BERT model as visualized in Figure 3.3. The model learn two extra embeddings that denotes whether a particular token is a part of history conversation or not. This introduces the conversation history to the BERT model in a natural way. This process improve the span of answer prediction and show considerable improvement in accuracy scores on QuAC's leader board. The authors, apart from introducing history answer embeddings, also tested their framework with three different experimental settings.

- **BERT:** This version of the conversation modeling is without any history modeling.
- **BERT+PHA:** This version prepends history answers in a BERT-based model.

- **BERT+PHQA:** This version prepends both history question and answers in a BERT-based model.

The experimentation results shows that introducing history answer embeddings perform considerably well rather than just prepending the history question and answers to the current question in a sequence.

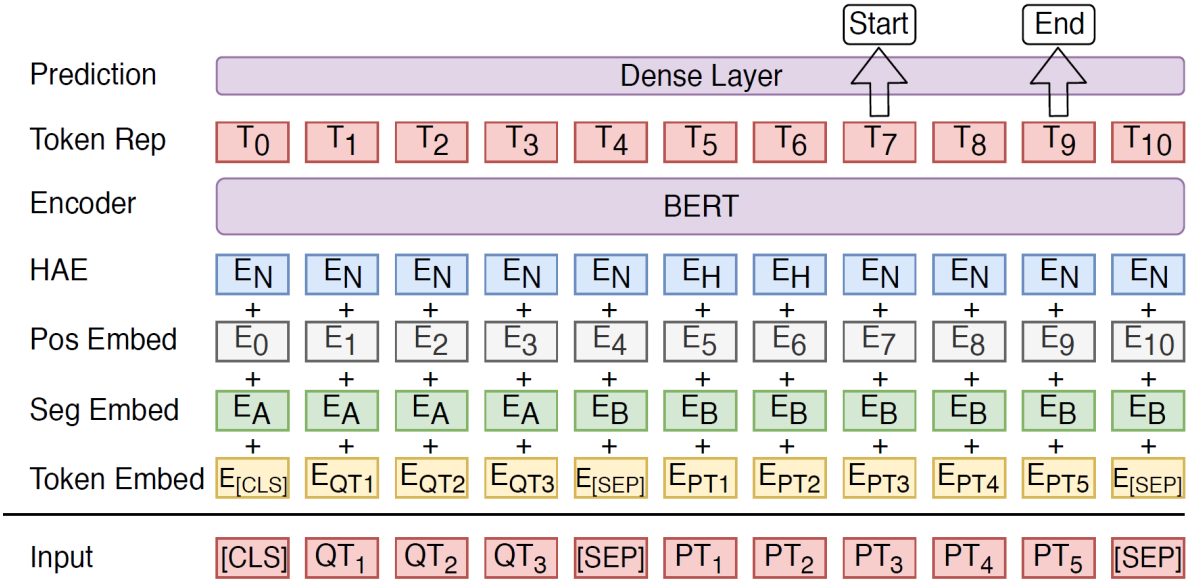


FIGURE 3.3: Architecture of the ConvQA model with HAE [42].

Another BERT based fine-tuning approach is introduced by Yasuhito et al. to improve the previously discussed models as represented in Figure 3.4. Their approach comprises of two steps. The first step is to generate contextual embeddings where BERT is utilized for independently getting context representations conditioned with the current query, previously asked queries, and each of the previous answers. The second step is answer prediction using span extraction. The start/end spans of the answer is predicted based on encoded representations in the previous step.

The authors conducted the experiments on two conversational machine comprehension datasets that are QuAC [11] and CoQA[46] and their method outperformed all the published models on the respective leader boards. In addition to this, they performed a detailed evaluation on the effect of varying number of turns and found out that gold answer history, which may not be present in the given context, added the most in improving the performance of both the models.

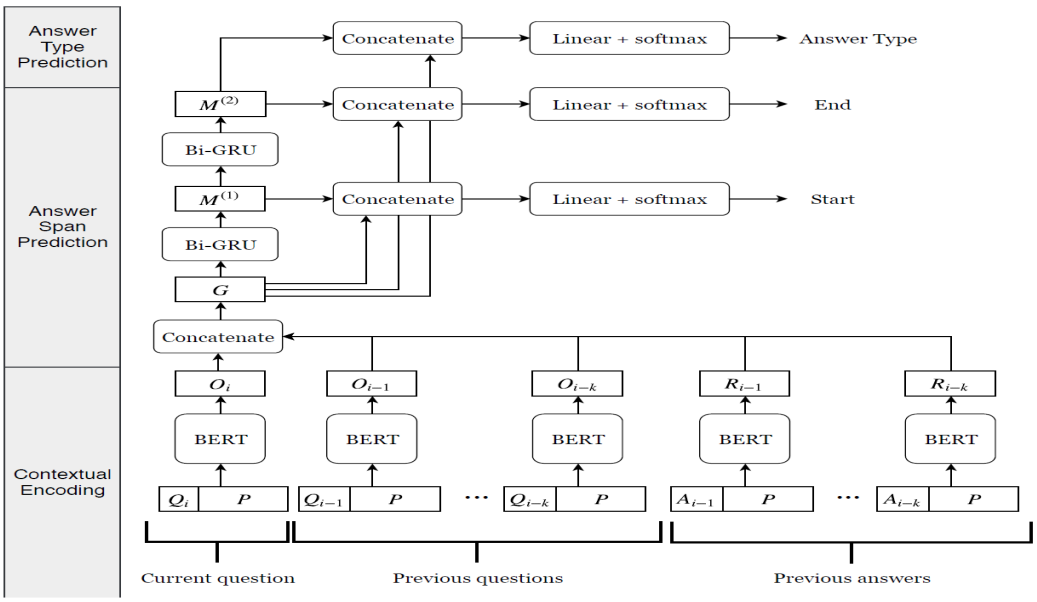


FIGURE 3.4: The general architecture of BERT with k/ctx [37].

3.3.1 Weaknesses

- Both BERT-based models achieved astonishing results owing to the bidirectional nature of the pre-trained language model. However, BERT-HAE worked on utilizing history answers, skipping the inclusion of history questions into the model.

- BERT w/k-ctx studied the effect of both history questions and answers but their approach lacked the mechanism of only extracting the history that is relevant to the current question. Using irrelevant history may bring in noise data.

3.4 Flow-Based CQA

Huang et al. propose an improved model, called FlowQA [22], to encode the history comprehensively. Their proposed mechanism incorporate intermediate representations generated during the process of answering the previous questions, through a parallel processing structure as shown in Figure 3.5. The advantage of this framework over others is that it integrates the latent semantics of the conversation history more deeply. When evaluated on the two conversational datasets and three domains of a sequential instruction understanding task, FlowQA outperforms all its competitors on the leader boards.

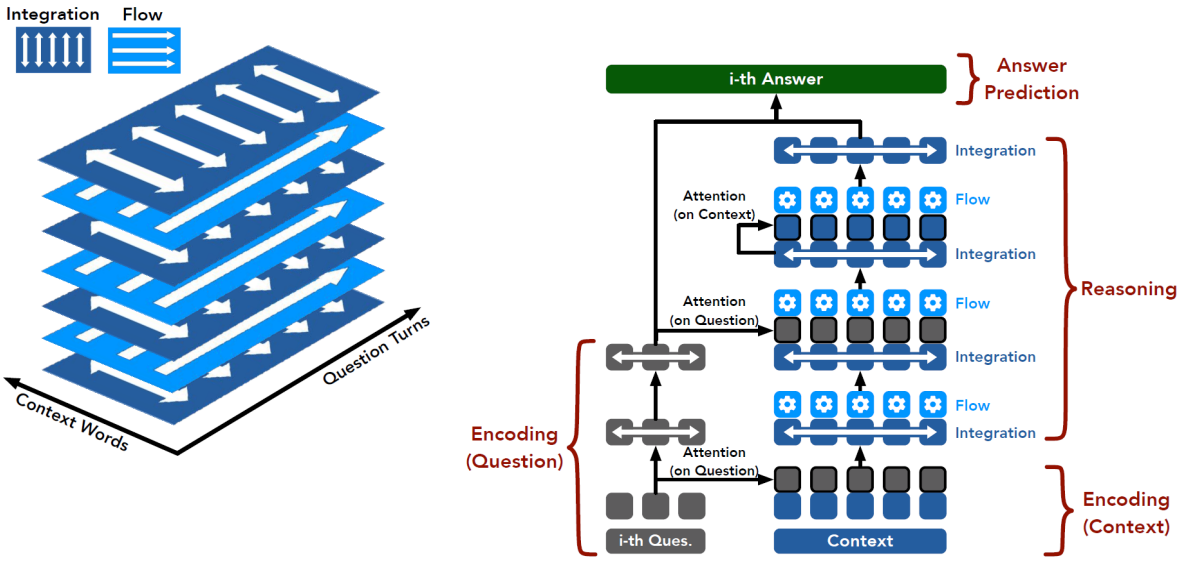


FIGURE 3.5: The general architecture of FlowQA [22]

3.4.1 Weaknesses

- The architecture proposed in FlowQA is complex to implement and creates computational overhead.

Figure 3.3, 3.4 shows that BERT-based architectures are comparatively easy to implement rather than the others (Figure 3.1, 3.2, 3.5) as all you need to do is finetuning of the model according to your task-specific needs. While the other models involve complex mechanism to generate contextual embeddings from the scratch and then integrate those into the respective architectures.

3.5 Conclusion

Although the above discussed models have contributed a lot for the progression of conversational machine comprehension, still there exists some loopholes in each approach. For example [49] and [12] do not consider modeling the history conversation when answering the current question. Bidaf++ w/k-ctx [11] models history but uses ELMo to generate the embeddings by employing a complex mechanism. Since ELMo is shallowly bidirectional, therefore, it does not capture the context of the words accurately. BERT-HAE [42] propose a very effective BERT-based approach but they only consider incorporating the history answer embeddings rather than considering both history questions and answers. Also, their history seletion module is rule-based i.e. you need to explicitly specify the number of turns to be considered. Thus, in this study we present a straight-forward but efficient approach to incorporate multi-turn dialogue using BERT for conversational machine comprehension.

Our method improves BERT-HAE by introducing both history question and answers to predict the start and end span of the answer to the current question. Also, we propose a dynamic history selection approach that computes the similarity score between the context paragraph and query, and selects the history turns that are relevant to the question.With the improvements introduced, our model addresses the issues discussed above and outperforms all the competing methods discussed in the next chapter.

4

Implementation Methodology

4.1 Overview

The research question of interest in this study is whether modeling the relevant conversation history into a BERT-based model would have an impact on the model's performance or not. The process of modeling conversation history includes selecting the relevant history turns using context-query relevance mechanism and then generating the history answer embeddings of the selected turns. These are provided as an input to the model along with prepended history questions to the given context paragraph. This study aims to determine how significant it is to model the relevant conversation history to improve the predicted answer span.

Our proposed framework is based on recently introduced pre-trained language model BERT hence we call our model **BERT-CoQAC (Conversational Question Answering in Context)**. Our approach leverage the language understanding capability of BERT to predict the answer to the current question provided the context paragraph. To predict the answer span correctly, we introduce a mechanism to model previous conversational history into the model as there is a probability that the premise of the current question might be dependent on the previously asked questions and their respective answers. To prove our hypothesis, we use a conversational question answering dataset,

QuAC. Furthermore, we evaluate our proposed model on a different number of turns without any history selection mechanism to examine the effect of a varying number of history turns on the accuracy of the answer.

This chapter will provide details on the task formulation, pre-trained model on which our framework is based, and proposed architecture. We, then, discuss the experimental setup that includes dataset description, and a brief description of competitive models. The chapter further sheds light on evaluation metrics and provide details on how our framework is implemented.

4.2 BERT-CoQAC Model

This section describes the details of our BERT-based Conversational Question Answering in Context model.

4.2.1 Task Formulation

Given a source paragraph and a question, the task is to find an answer to the question provided the context. If *P* denotes the given paragraph then the input and output of BERT-CoQAC can be formulated as follows:

- **Input:** The input of BERT-CoQAC consists of current question Q_i, given passage *P*, history questions $Q_{i-1}, ..., Q_{i-k}$ and the embedding of history answers $HE_{i-1}, ..., HE_{i-k}$,
- **Output:** The answer A_i identified using start span and end span generated by the model.

where *i* and *k* represent the indices of turn and the number of dialogue history considered, respectively.

4.2.2 Pre-trained Model

BERT is a newly launched language representation model and is based on a well-known Transformer encoder model [56]. It is a contextualized language model that is pre-trained on huge unlabeled corpora from Wikipedia and BookCorpus. With extra projection layers and fine-tuned deep structure, BERT has been successfully applied to various tasks such as reading comprehension, named entity recognition, and sentiment analysis. The general visualization of BERT is shown in Figure 4.1. Unlike other language models such as ELMo [40] or GPT-2 [44], BERT is designed to capture the context from both the directions in all the layers. As a result, an addition of just a single output layer can achieve outstanding results in a variety of NLP tasks. On the QuAC leaderboard[1], various approaches

[1] https://quac.ai/

[63, 24, 41] use BERT as a base model for conversational question answering including BERT-HAE [42]. The difference between BERT-HAE and our model is that they only capture the embeddings of all the previous history answers whereas our proposed BERT-CoQAC provides selected history questions with the embeddings of history answers as an input to the model. A particular input format is required

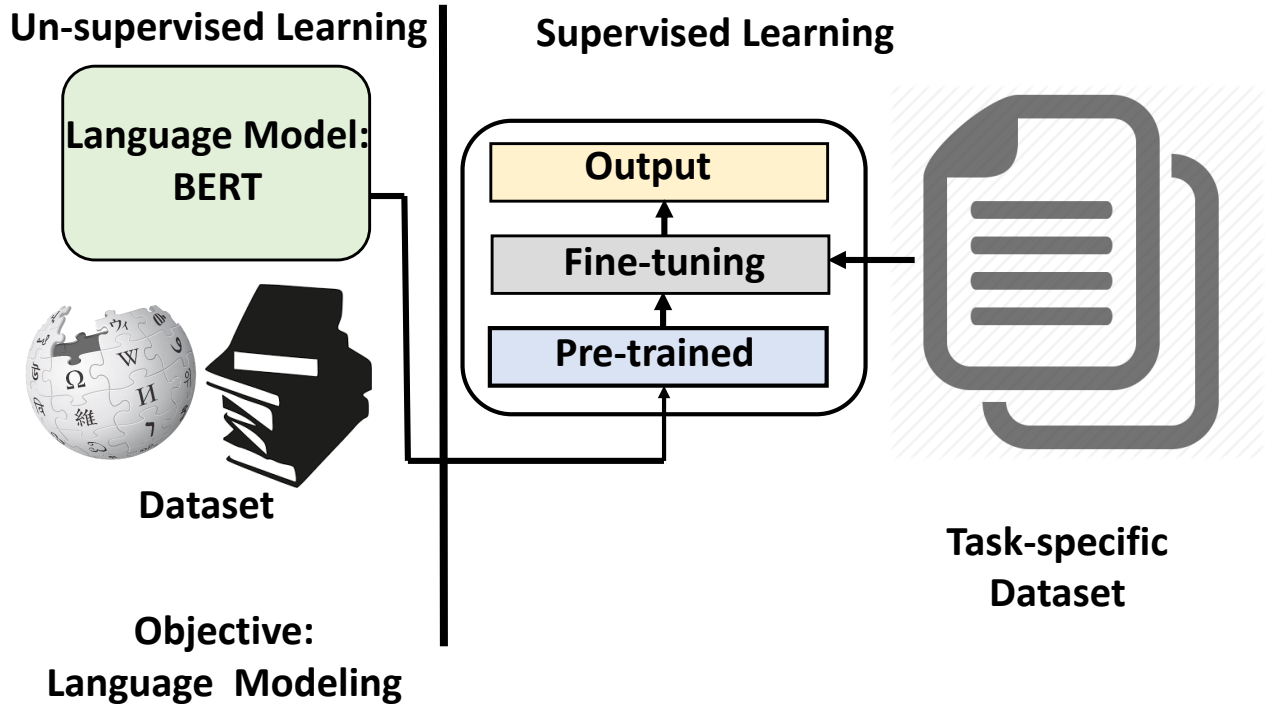


FIGURE 4.1: The high-level visualization of BERT model.

to make BERT adaptable to machine-comprehension task such as SQuAD [45]. In SQuAD, query and a context paragraph are given and the goal is to find the start and end span of the answer in the paragraph. To use BERT for SQuAD, a special classification token, [CLS], is added before the query. After that, another special token, [SEP] is added to concatenate query with the given paragraph to form a sequence. The sequence is provided as an input to the model along with positional embedding. Subsequently, the fully connected layer coupled with softmax activation function is utilized to predict the range of the answer. Let T_i be the BERT-representation of the i^{th} token and S be the initialization vector. The likelihood of the said token being the start token is $P_i = \frac{e^{S.T_i}}{\sum_k e^{S.T_k}}$. The probability of a token being the end token is computed in the same manner. The mean of the cross-entropy loss computed from the start and end positions is regarded as the loss. The fine-tuned BERT can effortlessly capture the relationship between a single question and the paragraph but has the limitation in capturing multi-turn questions in CMC tasks. This study is an effort to provide an efficient way of modeling history turns into a BERT-based model.

4.2.3 Modular Representation of BERT-CoQAC.

Figure 4.2 represents the modular framework of BERT-CoQAC, which consists of two main functions:

- The **History Selection Module** depends on explicitly calculating context-query relevance score that selects the history turns that are expected to be more relevant to the current question, and
- The **History Modeling Module** is embedded inside the BERT-CoQAC model. It takes the selected history conversation (both questions and answers) along with the current question and passage, and transforms them into the format required by BERT. The output of this module helps the model to predict the start and end span of the required answer.

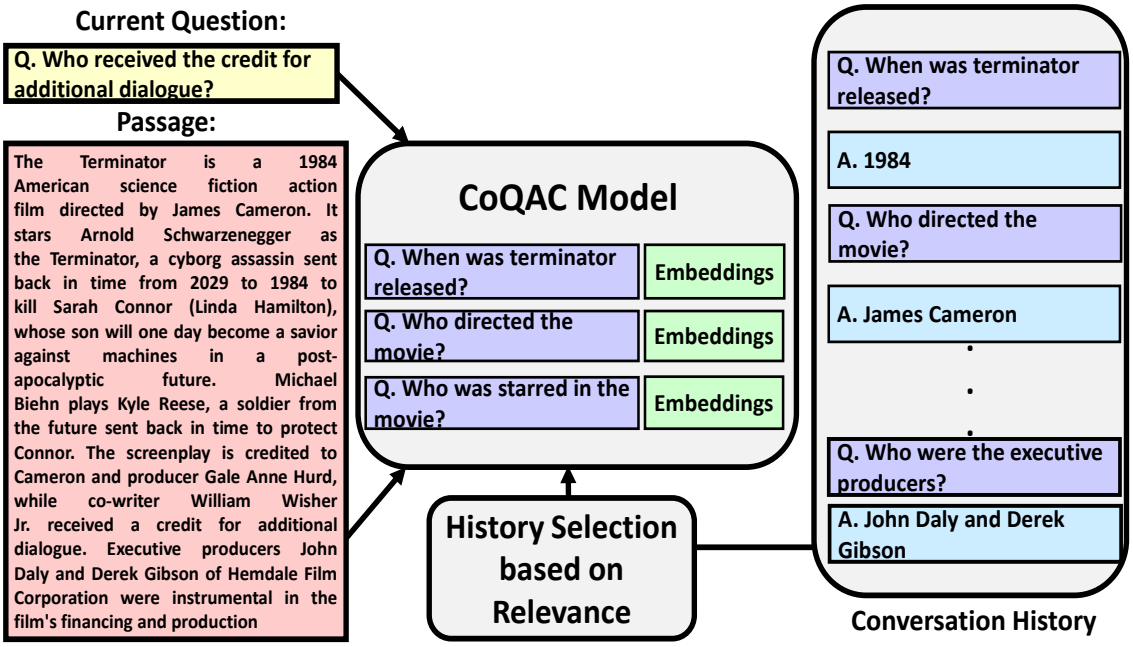


FIGURE 4.2: Modular representation of BERT-CoQAC. It shows the input formulation and the components of our model.

4.2.4 The BERT-CoQAC Architecture

In conversational question answering, it is essential to not only consider the current question and passage but also the previous questions and answers in order to understand the complete context of the conversation. There are a number of ways to embed the history turns into the system. The most common one is to prepend the previous question and answers to the current question [68, 44]. Another approach [42] is by providing an extra history answer embedding information to tokens. We leverage the strengths of pre-trained language model and adapt BERT to suit our task's requirements. Figure 4.3

illustrates our model's architecture that includes embeddings of history answers along with history questions. Another factor worth noting here is that only those history questions and answer embeddings are included that has higher cosine similarity score [2] calculated using

$$score_i = \frac{h_i.q_i}{||h_i||.||q_i||} \tag{4.1}$$

where h_i denotes current history turn and q represents the query. Finally the scores are normalized using softmax to get the probability distribution.

$$p_i = \frac{exp(score_i)}{\sum_{j=0}^{n} exp(score_i)} \tag{4.2}$$

The intuition behind context-query relevance is that, if the context is less relevant then we should not consider it as irrelevant information brings in the noise which deteriorates the accuracy of answer prediction. However, if the context is relevant, we should focus more evenly across the query and context. Thus, this model introduces the history turns to BERT in an efficient way. The model generates two types of tokens for history answer embedding that shows whether the token is part of history answer or not. These embeddings have influence on the information that the other tokens possess. The history questions are not part of the passage, so we cannot "embed" them directly to the input sequences. Instead, we prepend the historical questions to the context paragraph in the same sequence as that of embeddings to improve the answer span.

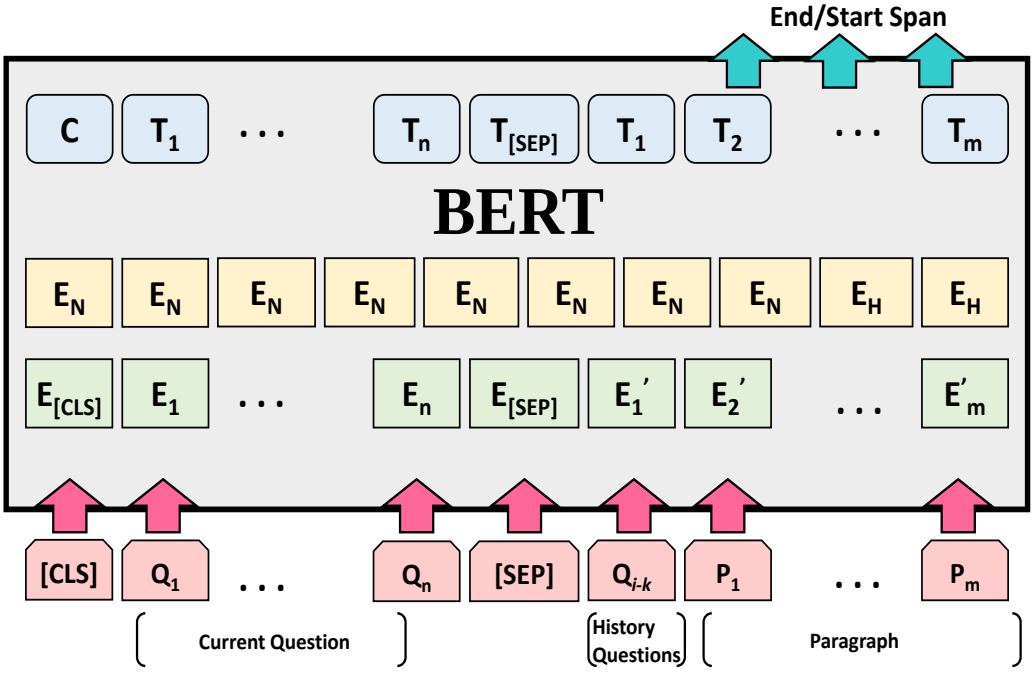


FIGURE 4.3: Architecture of BERT-CoQAC model. History questions are prepended with the passage and E_N/E_H denote whether the token is present in history or not.

4.2.5 Model Training

(Q_i, Q_i^k.P, HA_i^k, A_i) denotes a single instance of training where Q_i^k.*P* denotes the paragraph prepended with history questions in the same order as that of history answers, *HA*. This instance is first transformed into example variation where each variation has only one history turn (the last one) from the conversation history. The context-query relevance based history selection module then considers *k* selected history turns. A new instance, $(Q_i, Q_i^k.P, HE_i^k, A_i)'$, is formed by merging all the selected variations and used as an input to the BERT-CoQAC model. Since the length of the passages is greater than the max sequence length, therefore, we use the sliding window approach to split lengthy passages as suggested in the BERT [14] model.

4.3 Experimental Setup

In this section, we describe the setup of our experiments for the evaluation of the proposed BERT-CoQAC model, including the dataset, the comparison baseline methods, and the evaluation metrics.

4.3.1 The Dataset

High quality conversational dataset such as QuAC [11] and CoQA [46] have provided the researchers a great source to work deeply in the field of CMC. In our work, we choose to work on QuAC because it encourages users to participate more in the information seeking dialogue. The motivation behind QuAC (Question Answering in Context) comes from the idea of teacher-student setup where a student asks a series of questions about a topic to get in-depth information about it. The issue in this scenario is that most of the questions are context-dependant, can be abstract, and might not have any answer. It is up to the teacher to utilize and shift all the knowledge they possess to provide the best possible answer. The seeker can only see the title of the given paragraph and the answers are provided by generating the start and end span in the paragraph. Since it requires less background knowledge to ask questions about the people, so QuAC is comprised of articles from Wikipedia that are all about people. These people belong to a wide range of domains. QuAC is a unique dataset as it contains a wide variety of question types. It contains unanswerable questions. It is context-dependent. The comparison of QuAC with other question answering datasets is shown in Table 4.1. The training/validations sets have 11,000/1,000 questions across 14,000 dialogues. Every dialogue can have a maximum of 12 dialogue turns, which constitutes 11 history turns at most. Dialogue continue until all 12 questions were answered, two unanswerable questions are asked in a row, or one of the two people ends the dialogue manually. The dataset is very much unsolved. Human performance is at an F1 of 81.1 %, but

the top performing baseline is far behind this score. There is a room for a lot of progress to improve the accuracy score of this dataset.

Dataset	Multi-turn	Text-based	Dialog Acts	Simple Evaluation	Unanswerable Questions	Asker Can't See Evidence
QuAC [11]	✓	✓	✓	✓	✓	✓
CoQA[46]	✓	✓	✗	✓	✓	✗
CSQA[48]	✓	✗	✗	✗	✓	✗
CQA[52]	✓	✓	✗	✓	✗	✓
SQA[23]	✓	✗	✗	✓	✗	✗
NarrativeQA[27]	✗	✓	✗	✗	✗	✓
TriviaQA[25]	✗	✓	✗	✓	✗	✓
SQuAD 2.0[45]	✗	✓	✗	✓	✓	✗
MS Macro[36]	✗	✓	✗	✗	✓	✓
NewsQA[55]	✗	✓	✗	✓	✓	✓

TABLE 4.1: Comparison of the QuAC dataset to other Question Answering datasets [11].

4.3.2 Comparison Systems

We compare our model with the main baseline models and other recently published models on the QuAC leaderboard. A brief description of these methods is as follows:

- **BiDAF++ [49]:** BiDAF++ extends BiDAF by introducing contextualized embeddings and self-attention mechanism in the model.
- **BiDAF++ w/ 2-ctx [11]:** It takes 2 history turns into account when extending BiDAF++. It also concatenates the marker embeddings to passage embeddings and adds dialogue turn number into question embeddings.
- **FlowQA [22]:** FlowQA introduces a mechanism that incorporates intermediate representations generated during the process of answering previous questions to make the model be able to grasp the latent semantics of the history.
- **BERT-HAE [42]:** This model is built on pre-trained language model, BERT, to model history conversation using history answer embeddings. Similar to as that of BERT-HAE, we also compare our model with several variants.

- **BERT:** This version of the conversation modeling is based directly on BERT without any history modeling.
- **BERT+PHA:** This variant introduces the history answers as a part of history modeling to the model by prepending it to the given context.
- **BERT+PHQA:** This variant integrates the history modeling into the model by prepending both history question and answers to the context paragraph.

Our proposed BERT-CoQAC is an attempt to provide a dynamic history selection process that takes both history answer embeddings and history questions into consideration to improve the model's accuracy.

4.3.3 Evaluation Metrics

To evaluate our model, we use not only the F_1 score but also the human equivalence score for questions (HEQ-Q) and human equivalence score for dialogues (HEQ-D) [11]. HEQ-Q represents the percentage of exceeding or matching the human performance on questions and HEQ-D represents the percentage of exceeding or matching the human performance on dialogues.

4.3.4 Implementation Details:

We implement BERT-base-uncased Tensorflow model released by Google [2]. We set maximum sequence length to 384, batch size to 12, and maximum query length is set to 64. Document stride is 128 and the maximum answer length is of 40. The optimizer is Adam with a learning rate of 3e-5. The number of training epochs is 3. We use two NVIDIA Tesla P100 16GB GPUs. We test our model on the validation set with maximum history turns possible (i.e. 11) as shown in Table 5.2. For the dataset, we use QuAC V0.2.

4.4 Conclusion

This chapter has provided an overview about the technical details of our proposed model, BERT-CoQAC. The chapter covered the information about dataset used, the modular breakdown of our framework, how history turns are introduced into the model, and evaluation metrics used. The results of the experiments conducted is discussed in next chapter.

[2] https://github.com/google-research/bert

5

Evaluation Results

5.1 Overview

This section presents the significant results that we have obtained by performing extensive experiments using our framework, BERT-CoQAC. First of all, we model the selected conversation history into our BERT-based model and compare our results with that of leading models on QuAC leaderboard [1]. Then, we perform extensive ablation experiments in order to understand the importance of history selection mechanism to improve the answer span prediction.

5.2 Evaluation Analysis

Table 5.1 demonstrates the evaluation outcomes of our model, BERT-CoQAC, on the QuAC dataset. Our model outperforms the baseline methods and BERT-HAE model on all the three metrics i.e. F_1, HEQ-Q and, HEQ-D. From the results, we can make the following observations:

- Using conversation history has a significant effect when answering the current question. This holds true for both BiDAF++ and BERT-based methods.

[1] http://quac.ai/

- Our history modeling technique is better than the baseline methods and outperforms BiDAF++ w/2-ctx with reasonably good margin.
- Since BERT is a pre-trained contextualized model therefore any BERT-based model would perform better than the current baseline methods and are confirmed by our experimental results. This demonstrates the leverage of using BERT for conversational systems.
- Incorporating relevant history is essential for a model to understand the context of the current question more accurately. Our model outperforms BERT-HAE that uses only history answer embeddings in their model. Also, Table 5.1 shows that our model gives better performance by using a dynamic history turns selection mechanism.
- BERT-CoQAC performs comparatively well with simple experiment setup as compared to FlowQA that uses convoluted mechanisms to model the history.
- Apart from comparing the model's efficiency, we also calculate the computational efficiency. We find that our model takes at least 5 times lesser time than FlowQA [22] when training. Thus, our model achieves comparable performance to FlowQA with much lesser training time and computation efficiency.

Models	F1	HEQ-Q	HEQ-D	Train Time (h)
BiDAF++	51.8/50.2	45.3/43.3	2.0/2.2	-
BiDAF++ w/2Context	60.6/60.1	55.7/54.8	5.3/4.0	-
BERT	54.4/-	48.9/-	2.9/-	-
BERT + PHA	61.8/-	57.5/-	4.7/-	-
BERT + PHQA	62.0/-	57.5/-	5.4/-	-
BERT + HAE	63.1/62.4	58.6/57.8	6.0/5.1	-
BERT-CoQAC	**64.24/63.2**	**59.1/60.0**	**6.3/5.5**	**10.1**
FlowQA	-/64.1	-/59.6	-/5.8	-/56.8

TABLE 5.1: The evaluation results on the test set of QuAC dataset. The top section is the baseline methods, the middle section is BERT-HAE with different methods and the bottom section lists the best performing model.

5.2.1 Ablation Studies

To understand the significance of history selection mechanism and prepending history questions along with the history answer embeddings, we created a similar experimental setup as BERT-HAE [42]. Their method uses rule-based history selection which selects the previous *j* turns and is based on the perception that the closer closer conversational history retains more relevant information to the current question. The results of our experiments are as follows:

- Table 5.2 shows the F_1, HEQ-Q and, HEQ-D scores from our experiments on different number of history turns. From the results we can see that the overall accuracy is marginally low when no

Evaluation with 11 history turns on QuAC			
History Turns	**F1**	**HEQ-Q**	**HEQ-D**
1	61.57	57.58	4.7
2	63.04	58.9	6.0
3	62.58	58.64	5.4
4	62.46	58.04	5.4
5	**63.4**	**58.86**	**6.3**
6	62.73	58.39	5.8
7	62.94	58.89	6.2
8	62.16	58.10	4.6
9	62.9	58.05	5.6
10	62.23	58.26	5.7
11	62.13	58.11	5.6

TABLE 5.2: The evaluation results of BERT-CoQAC with the varying number of turns on the development set of QuAC.

history selection mechanism is used.

- Figure 5.1 shows that it is effective to use the previous turns based on both questions and answers to improve the accuracy. Our ablated model still achieve better results than the BERT-HAE model just by introducing history questions into the model.
- Our model achieves higher score using 5 turns as shown in Figure 5.2 (the next best accuracy is with 2 turns) which is an improvement over BERT-HAE that obtained highest accuracy in step 6.
- Figure 5.2 compare our model's performance with human performance and the BERT-HAE [42] model. Since human performance was evaluated on 10 turns, therefore, we have used 10 turns for both the BERT-HAE and BERT-CoQAC models. The highest accuracy for human evaluation is 81.1% in case of F_1 score. Our model provides more accuracy than BERT-HAE at 5 different points. Also for HEQ-D, BERT-CoQAC produces better results than the BERT-HAE model. The human evaluation score is 100% for HEQ-D.

5.3 Conclusion

The obtained results confirm our hypothesis that it is significant to have a dynamic history selection mechanism to only consider the context that is relevant to the current question. It can also be concluded from our results that modeling both previous questions and answers is effective to provide a better understanding of the question as it helps to solve the co-references used in the current question. Thus, we can safely say that our model provides a simple but effective way to incorporate the relevant history turns into conversational machine comprehension systems. The next chapter concludes our work and

provide insights and possibilities for future direction.

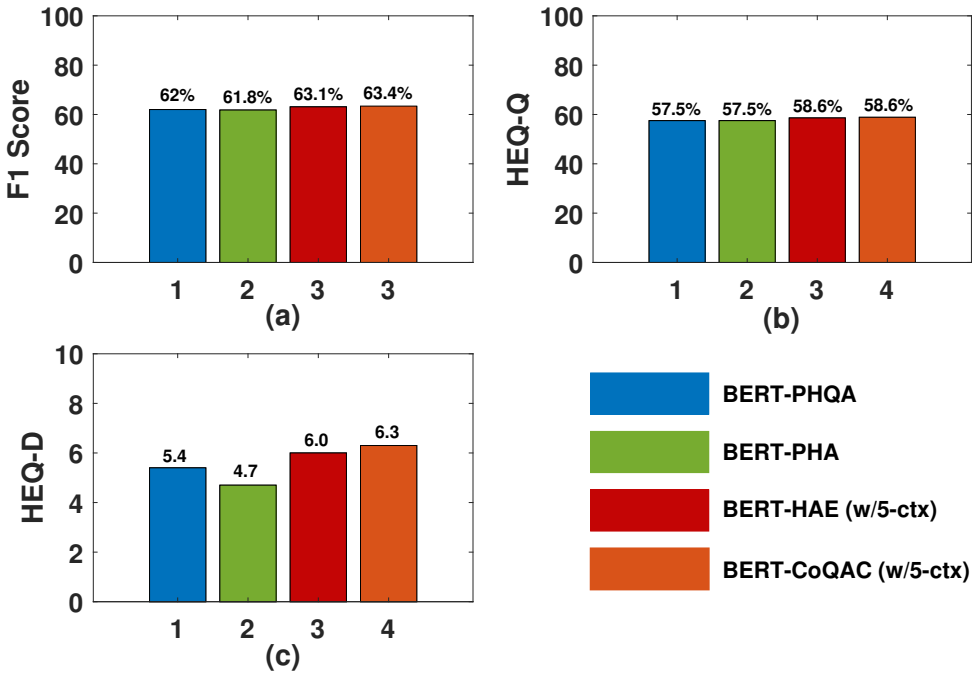
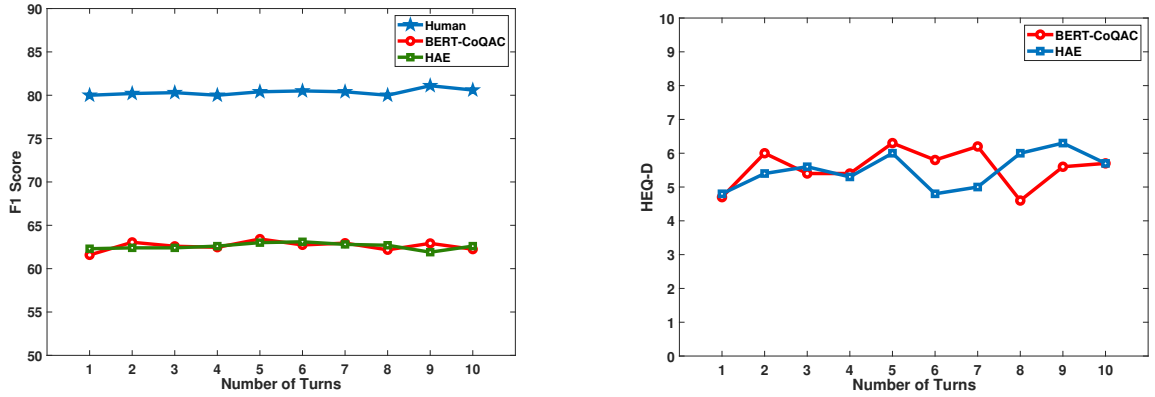


FIGURE 5.1: Ablation over history questions. Comparison of our ablated model with BERT-HAE and it's variants.



(a) The F_1 scores with different number of turns on QuAC development set.

(b) The HEQ-D score with different number of turns on QuAC developement set.

FIGURE 5.2: Evaluation Results

6

Conclusion and Future Work

6.1 Overview

The pre-trained language models promise to alleviate the long-standing problem of data scarcity and bring about remarkable improvements in natural language understanding and generation. Although the use of these pre-trained models in machine comprehension has gained a lot of interest from both industry researchers and academicians, research on their history modeling aspect is still very limited. In this thesis, we have made history modeling in conversational machine comprehension as our point of focus. We have focused on using BERT to generate the contextual embeddings for our model and demonstrate how the strengths of this powerful model can be leveraged to model history when answering the current question. In this chapter, summarize the research work and highlight our main contributions. We also discuss the probable future directions to extend the current research.

6.2 Conclusion

This thesis has discussed machine comprehension which can be classified as single-turn machine comprehension or multi-turn machine comprehension. The main focus of our study is multi-turn

machine comprehension. In multi-turn machine comprehension, also known as conversational machine comprehension or sequential machine comprehension, the asker asks a series of questions and the adequate modeling of the history questions is very crucial to understand the context of the current question.

In this study, we have introduced a BERT-based framework, BERT-CoQAC, for effective conversational question answering in context. BERT-CoQAC first selects the relevant conversational history to be fed as an input and then extends the BERT-HAE model to add the history questions along with the embeddings to generate better context of the conversation. The experimentation results demonstrates the effectiveness of our approach and addresses the shortcomings of the previous works. To verify our hypothesis, we conducted ablation studies as well and performed a number of experiments to analyze the effectiveness of our approach on a real-world dataset.

6.3 Future Work

Since this thesis represents only the preliminary work to undertake the Ph.D. studies, the study is far from complete.

In particular, our future work will consider to introduce the embeddings of history questions, replacing the history questions in textual form, to analyze its effect on the model. Although our history selection mechanism contributed in improving the accuracy but it is done on a very basic level. This process has still a lot of room for improvisation as future work. The other future research plan is to introduce an advanced strategy to select the most relevant history turns by analyzing the relationship between history turns and dynamically assigning a weightage.

In general, most of the machine comprehension systems are based on semantic matching i.e relevance between the previous context and the current question is calculated to answer the question, which makes the systems incapable of reasoning. This lack of inference capability often results in inaccurate answers. Thus, much effort is required in this area to provide strong reasoning skills to the such systems. Also, in question answering datasets, there are certain questions whose answers are not directly implied. In this scenario, human intelligence would utilize some external knowledge or common sense to find answers to the questions. To imitate this behaviour knowledge based machine comprehension is introduced but how to effectively introduce and integrate the external knowledge is still an open challenge. In the future Ph.D. tenure, we plan to address the above mentioned issues to make our contribution in the field of machine comprehension.

Bibliography

[1] Gerard Briones, Kasun Amarasinghe, and Bridget T. McInnes. “VCU-TSA at Semeval-2016 Task 4: Sentiment Analysis in Twitter”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*. 2016, pp. 215–219.

[2] Akshay Chaturvedi, Onkar Arun Pandit, and Utpal Garain. “CNN for Text-Based Multiple Choice Question Answering”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*. 2018, pp. 272–277.

[3] Danqi Chen, Jason Bolton, and Christopher D. Manning. “A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. 2016.

[4] Danqi Chen et al. “Reading Wikipedia to Answer Open-Domain Questions”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. 2017, pp. 1870–1879.

[5] Hongshen Chen et al. “A Survey on Dialogue Systems: Recent Advances and New Frontiers”. In: *SIGKDD Explorations* 19.2 (2017), pp. 25–35.

[6] Yu Chen, Lingfei Wu, and Mohammed J. Zaki. “GraphFlow: Exploiting Conversation Flow with Graph Neural Networks for Conversational Machine Comprehension”. In: *CoRR* abs/1908.00059 (2019).

[7] Zheqian Chen et al. “Smarnet: Teaching Machines to Read and Comprehend Like Human”. In: *ArXiv* abs/1710.02772 (2017).

[8] Zhipeng Chen et al. “Convolutional Spatial Attention Model for Reading Comprehension with Multiple-Choice Questions”. In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. 2019, pp. 6276–6283.

[9] Kyunghyun Cho et al. “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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. 2014, pp. 1724–1734.

[10] Eunsol Choi et al., eds. *Proceedings of the Workshop on Machine Reading for Question Answering@ACL 2018, Melbourne, Australia, July 19, 2018*. Association for Computational Linguistics, 2018. ISBN: 978-1-948087-39-1.

[11] Eunsol Choi et al. “QuAC: Question Answering in Context”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. 2018, pp. 2174–2184.

[12] Christopher Clark and Matt Gardner. “Simple and Effective Multi-Paragraph Reading Comprehension”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. 2018, pp. 845–855.

[13] Yiming Cui et al. “Attention-over-Attention Neural Networks for Reading Comprehension”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. 2017, pp. 593–602.

[14] Jacob Devlin et al. “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 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. 2019, pp. 4171–4186.

[15] Bhuwan Dhingra et al. “A Comparative Study of Word Embeddings for Reading Comprehension”. In: *CoRR* abs/1703.00993 (2017).

[16] Bhuwan Dhingra et al. “Gated-Attention Readers for Text Comprehension”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. 2017, pp. 1832–1846.

[17] Ian J. Goodfellow et al. “Maxout Networks”. In: *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*. 2013, pp. 1319–1327.

[18] Wei He et al. “DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications”. In: *Proceedings of the Workshop on Machine Reading for Question Answering@ACL 2018, Melbourne, Australia, July 19, 2018*. Ed. by Eunsol Choi et al. Association for Computational Linguistics, 2018, pp. 37–46. ISBN: 978-1-948087-39-1.

[19] Karl Moritz Hermann et al. “Teaching Machines to Read and Comprehend”. In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 2015, pp. 1693–1701.

[20] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[21] Minghao Hu et al. “Reinforced Mnemonic Reader for Machine Reading Comprehension”. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.* 2018, pp. 4099–4106.

[22] Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. “FlowQA: Grasping Flow in History for Conversational Machine Comprehension”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019.

[23] Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. “Search-based Neural Structured Learning for Sequential Question Answering”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. 2017, pp. 1821–1831.

[24] Chanwoo Jeong et al. “A Context-Aware Citation Recommendation Model with BERT and Graph Convolutional Networks”. In: *CoRR* abs/1903.06464 (2019).

[25] Mandar Joshi et al. “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, July 2017, pp. 1601–1611.

[26] Rudolf Kadlec et al. “Text Understanding with the Attention Sum Reader Network”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. 2016.

[27] Tomáš Kočiský et al. “The NarrativeQA Reading Comprehension Challenge”. In: *Transactions of the Association for Computational Linguistics* 6 (2018), pp. 317–328.

[28] Guokun Lai et al. “RACE: Large-scale ReAding Comprehension Dataset From Examinations”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. 2017, pp. 785–794.

[29] Shanshan Liu et al. “Neural Machine Reading Comprehension: Methods and Trends”. In: *CoRR* abs/1907.01118 (2019). arXiv: 1907.01118. URL: http://arxiv.org/abs/1907.01118.

[30] Thang Luong, Hieu Pham, and Christopher D. Manning. “Effective Approaches to Attention-based Neural Machine Translation”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. 2015, pp. 1412–1421.

[31] Kaixin Ma, Tomasz Jurczyk, and Jinho D. Choi. “Challenging Reading Comprehension on Daily Conversation: Passage Completion on Multiparty Dialog”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. 2018, pp. 2039–2048.

[32] Bryan McCann et al. “Learned in Translation: Contextualized Word Vectors”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 2017, pp. 6294–6305.

[33] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and Their Compositionality”. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’13. Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 3111–3119.

[34] Sewon Min et al. “Efficient and Robust Question Answering from Minimal Context over Documents”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. 2018, pp. 1725–1735.

[35] Ramesh Nallapati et al. “Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond”. In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 280–290.

[36] Tri Nguyen et al. “MS MARCO: A Human Generated MAchine Reading COmprehension Dataset”. In: *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*. 2016.

[37] Yasuhito Ohsugi et al. “A Simple but Effective Method to Incorporate Multi-turn Context with BERT for Conversational Machine Comprehension”. In: *CoRR* abs/1905.12848 (2019).

[38] Boyuan Pan et al. “Memen: Multi-layer embedding with memory networks for machine comprehension”. In: *arXiv preprint arXiv:1707.09098* (2017).

[39] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. “Glove: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. 2014, pp. 1532–1543.

[40] Matthew E. Peters et al. “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. 2018, pp. 2227–2237.

[41] C. Qu et al. “Attentive History Selection for Conversational Question Answering”. In: *CIKM ’19*. 2019.

[42] Chen Qu et al. “BERT with History Answer Embedding for Conversational Question Answering”. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. 2019, pp. 1133–1136.

[43] Alec Radford et al. “Improving language understanding by generative pre-training”. In: *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf* (2018).

[44] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI Blog* 1.8 (2019).

[45] Pranav Rajpurkar et al. “SQuAD: 100, 000+ Questions for Machine Comprehension of Text”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. 2016, pp. 2383–2392.

[46] Siva Reddy, Danqi Chen, and Christopher D. Manning. “CoQA: A Conversational Question Answering Challenge”. In: *TACL* 7 (2019), pp. 249–266.

[47] Matthew Richardson, Christopher J. C. 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, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. 2013, pp. 193–203.

[48] Amrita Saha et al. "Complex Sequential Question Answering: Towards Learning to Converse Over Linked Question Answer Pairs with a Knowledge Graph". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 2018, pp. 705–713.

[49] Min Joon Seo et al. "Bidirectional Attention Flow for Machine Comprehension". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017.

[50] Alessandro Sordoni, Philip Bachman, and Yoshua Bengio. "Iterative Alternating Neural Attention for Machine Reading". In: *CoRR* abs/1606.02245 (2016).

[51] Sainbayar Sukhbaatar et al. "End-To-End Memory Networks". In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 2015, pp. 2440–2448.

[52] Alon Talmor et al. "CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. 2019, pp. 4149–4158.

[53] Chuanqi Tan et al. "S-Net: From Answer Extraction to Answer Generation for Machine Reading Comprehension". In: *CoRR* abs/1706.04815 (2017).

[54] Adam Trischler et al. "Natural Language Comprehension with the EpiReader". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. 2016, pp. 128–137.

[55] Adam Trischler et al. "NewsQA: A Machine Comprehension Dataset". In: *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*. 2017, pp. 191–200.

[56] Ashish Vaswani et al. “Attention is All You Need”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 978-1-5108-6096-4.

[57] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. “Pointer Networks”. In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 2015, pp. 2692–2700.

[58] Shuohang Wang and Jing Jiang. “Machine Comprehension Using Match-LSTM and Answer Pointer”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017.

[59] Wenhui Wang et al. “Gated Self-Matching Networks for Reading Comprehension and Question Answering”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. 2017, pp. 189–198.

[60] Jason Weston, Sumit Chopra, and Antoine Bordes. “Memory Networks”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.

[61] Caiming Xiong, Victor Zhong, and Richard Socher. “DCN+: Mixed Objective And Deep Residual Coattention for Question Answering”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 2018.

[62] Caiming Xiong, Victor Zhong, and Richard Socher. “Dynamic Coattention Networks For Question Answering”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017.

[63] Yi Ting Yeh and Yun-Nung Chen. “FlowDelta: Modeling Flow Information Gain in Reasoning for Conversational Machine Comprehension”. In: *CoRR* abs/1908.05117 (2019).

[64] Adams Wei Yu et al. “QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 2018.

[65] Yang Yu et al. “End-to-end answer chunk extraction and ranking for reading comprehension”. In: *arXiv preprint arXiv:1610.09996* (2016).

[66] Junbei Zhang et al. “Exploring Question Understanding and Adaptation in Neural-Network-Based Question Answering”. In: *CoRR* abs/1703.04617 (2017).

[67] Li Zhou et al. “The Design and Implementation of XiaoIce, an Empathetic Social Chatbot”. In: *CoRR* abs/1812.08989 (2018).

[68] Chenguang Zhu, Michael Zeng, and Xuedong Huang. “SDNet: Contextualized Attention-based Deep Network for Conversational Question Answering”. In: *CoRR* abs/1812.03593 (2018).

[69] Haichao Zhu et al. “Hierarchical Attention Flow for Multiple-Choice Reading Comprehension”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 2018, pp. 6077–6085.