HIERARCHICAL ATTENTION NETWORK ARCHITECTURE IMPROVEMENT FOR CLOZE-STYLE QUESTION ANSWERING

BY

FAHAD ALSAHLI

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

INFORMATION AND COMPUTER SCIENCES

DECEMBER, 2020

HIERARCHICAL ATTENTION NETWORK ARCHITECTURE IMPROVEMENT FOR CLOZE-STYLE QUESTION ANSWERING

BY

FAHAD ALSAHLI

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

In Partial Fulfillment of the Requirements
for the Degree of

MASTER OF SCIENCE

IN

INFORMATION AND COMPUTER SCIENCES

KING FAHD UNIVERSITY
OF PETROLEUM & MINERALS

Dhahran, Saudi Arabia

DECEMBER, 2020

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **FAHAD ALSAHLI** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN INFORMATION AND COMPUTER SCIENCES.**

Thesis Committee

Dr. Andri Mirzal (Adviser)

Dr. Hasan Luqman (Member)

Dr. Irfan Ahmad (Member)

Dr. Hamoud Aljamaan
Department Chairman

Dr. Suliman S. Al-Homidan
Dean of Graduate Studies

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Date

This study is dedicated to my family, friends, and colleagues who have helped and supported me during the study.

ACKNOWLEDGMENTS

I thank King Fahd University of Petroleum and Minerals for supporting my research. I would like to acknowledge Dr. Andri Mirzal for his help and support throughout the course of this research project. Also, I thank Dr. Hamzah Luqman, Dr. Irfan Ahmad, and Dr. Hamoud Aljamaan for their valuable advice.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

THESIS ABSTRACT

NAME: Fahad Alsahli

TITLE OF STUDY: Hierarchical Attention Network Architecture Improvement for Cloze-Style Question Answering

MAJOR FIELD: Information and Computer Sciences

DATE OF DEGREE: December, 2020

Recently, researchers have been addressing Question Answering (QA) by utilizing deep learning architectures, e.g., Recurrent Neural Networks (RNNs), Convolutional Neural Networks, and attention mechanism. QA has several variants, for example, document-based QA and cloze-style QA. In general, QA tasks could be addressed via similar approaches. This is due to the nature of QA which needs a context and a question to be analyzed so that an answer can be retrieved. In this thesis, we are tackling cloze-style QA. In such tasks, a context and a query are given. Query is a sentence that is missing a piece of information (e.g., a word). The missing information should be inferred based on the given context. Hierarchical Attention Network (HAN) based models employ hierarchical attention, so they are suitable for cloze-style QA task. HAN models have two layers of text encoding and use Global Vectors (Glove) embed-

dings. We propose a HAN model to address cloze-style QA. Our proposed HAN model has a single layer of text encoding. We conduct experiments to compare proposed model against a baseline model (HAN pointer sum attention) having two layers of text encoding. Comparison is based on inference times (i.e., time needed to process and answer a sample) and accuracy scores. We utilize two publicly available cloze-style datasets which are two instances of Children's Book Test (CBT), namely, Named Entity (CBT-NE) and Common Nouns (CBT-CN). We also use simplified versions of Cable News Network (CNN) and Daily Mail data. Results show that proposed model has lower inference time compared to baseline while maintaining baseline's accuracy score. Proposed model achieves an average inference time reduction of 41.39%. Moreover, we investigate effects of different Bidirectional Encoder Representations from Transformers (BERT) embeddings on accuracy score of HAN. We find that embeddings extracted from BERT's first layer give the best accuracy score compared to embeddings extracted from other layers. Then, we design and run experiments to compare BERT's first layer embeddings against Glove embeddings. Experiments show that the two techniques of embeddings result in similar accuracy scores.

ملخص الرسالة

الاسم: فهد السهلي

عنوان الدراسة: تحسين أداء برامج إجابة الأسئلة باستخدام Hierarchical Attention Network

التخصص: علوم الحاسب و المعلومات

تاريخ الدرجة العلمية: 12-2020

يوجد العديد من مجالات معالجة اللغات الطبيعية. أحد هذه المجالات هو إجابة الأسئلة المتعلقة بموضوع معين. يهتم هذا المجال ببناء برامج تقوم بقراءة نصوص في موضوع معين، و تجيب عن أسئلة المهتمين بالموضوع بناءً على ما تمت قراءته. لتحسين أداء هذه البرامج، نماذج التعلم العميق أصبحت تستخدم في عملية بناء برامج إجابة الأسئلة. أحد هذه النماذج هو Hierarchical Attention Network. عملنا في هذه الدراسة على تحسين أداء النموذج من خلال تقليص الوقت اللَازم لمعالجة المدخلات و دراسة رفع دقة النموذج. استخدمنا أربع مجموعات مختلفة من البيانات خلال هذه الدراسة. عملية تحسين النموذج تمت على مرحلتين. المرحلة الأولى كانت عبارة عن تقليص عدد وحدات معالجة النصوص (Text Encoding Layers). نتج عن هذا التعديل انخفاض 41.39% في وقت معالجة النصوص مع عدم التأثير على دقة النموذج. بالنسبة للمرحلة الثانية، فهي عبارة عن دراسة تأثير Bidirectional Encoder Representations from Transformers (BERT) embeddings على دقة النموذج مقارنة بالطريقة التقليدية، وهي Global Vectors embeddings. توصلت المرحلة الثانية إلى أن BERT embeddings لا تحسن أداء النموذج مقارنة بالطريقة التقليدية.

CHAPTER 1

INTRODUCTION

The main concern of Natural Language Processing (NLP) is to represent semantics and syntactics of a natural language to a computer program. NLP is important because it automates tasks that are time consuming for humans such as searching or translating documents. NLP tasks could be simple such as finding stop words or extracting numbers from text because such tasks do not require semantic processing. NLP tasks could be moderate such as suggesting a better phrasing of a sentence or finding grammatical mistakes since these tasks do not need to understand the whole document. Finally, NLP problems could be complex as in the case of providing a summary of a document or answering a question based on a context within a document. Such problems need to process documents and understand interactions between words and sentences. Complex NLP problems are currently being solved via applying deep learning architectures such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks. One NLP task that is recently being solved with deep learning is Question Answering (QA).

QA is where a computer program (or a system) is given a question, and the task is to provide an answer. There are several variants of such programs. For example, there exist systems that provide answers based on Knowledge Bases. As the name implies, these systems analyze Knowledge Bases to provide users with answers. Some other systems are given questions paired with corresponding input documents, and their task is to answer questions based on the input documents.

Moreover, QA programs could be classified by the types of questions they are answering. For instance, some programs answer multiple choice questions. In this type of QA, programs are given questions, contexts, sets of possible answers. The goal of programs is to process contexts and pick the right answers. Other QA programs consider short-answer questions. To solve this task, a program is presented with a context and a question. Then, it needs to provide a short answer (e.g., a sentence) from the context. Also, there are systems that answer cloze-style questions. In such systems, a context and a sentence with missing information (e.g., a name) are given, and the task is to provide the missing piece of information. In cloze-style QA, the missing information is a single word [3], [4]. The following is a simple cloze-style example with a three-sentence context followed by a cloze sentence:

1. *Cryptography is a field that aims to develop secure encryption algorithms.*
2. *Secure encryption algorithms are needed to provide safe communications over the internet.*
3. *This is because the internet is not secure to communicate private information.*

4. *When a person wants to send sensitive information over the internet, they need to use a secure XXXXX.*

In the above example, XXXXX represents the missing information, and it should be replaced with the correct word which is, based on the context, algorithm. A set of approaches addressing different QA variants are discussed in chapter 3.

All mentioned classes of QA systems utilize similar deep learning techniques. This is due to the nature of the problem. For instance, every QA system should analyze a question based on a context then provide an answer. However, transforming a model from one QA task to another is not straightforward and requires some changes, especially on the input and output layers, [5]. As a result, we limit our research to cloze-style QA as it has several datasets, and each set has more than 100,000 training samples as shown in table 3.1. Hierarchical Attention Network (HAN) architecture [1] is suitable for cloze-style QA since HAN based models as it uses a hierarchical attention to learn semantics and syntactics properties of training data.

We improve HAN based model to address cloze-style QA tasks. There are several aspects of HAN architecture that could be improved. HAN models use two layers of recurrent text encoding which could be reduced to one layer. We argue that one layer of text encoding is sufficient for a HAN model to learn interactions of words in a given domain such as cloze-style QA. This is because HAN uses pre-trained Global Vectors (Glove) embeddings [6] which already capture meaning of words. Also, having a single

layer of text encoding would improve inference time (i.e., time to process and answer a single sample) as recurrent text encoding is time consuming since they process text sequentially. In addition, accuracy scores of HAN architecture might be improved by investigating the use of pre-trained Bidirectional Encoder Representations from Transformers (BERT) [7]. BERT provides better embeddings than Glove in many NLP tasks [7]. Thus, utilizing pre-trained BERT would be a better approach. In this research, we aim to improve HAN models for cloze-style QA by studying effects of text encoding layers and BERT embeddings. The following are the research questions that will be addressed in this research:

- Q1: What effect does one-layer text encoding have on HAN's inference time compared to two-layer text encoding?
- Q2: What effect does one-layer text encoding have on HAN's accuracy score compared to two-layer text encoding?
- Q3: How to improve HAN models, in terms of accuracy, by utilizing pre-trained BERT embeddings?

Corresponding hypotheses are formalized as:

- H_{01}: One-layer text encoding HAN model and two-layer text encoding HAN model have the same average inference time.
- H_{11}: One-layer text encoding HAN model and two-layer text encoding HAN model have different average inference times.

- H_{02}: One-layer text encoding HAN model and two-layer text encoding HAN model have the same average accuracy score.

- H_{12}: One-layer text encoding HAN model and two-layer text encoding HAN model have different average accuracy scores.

For the third question, a precise hypothesis cannot be formalized. This is due to the fact that there are many ways to extract embeddings from BERT [7]. It is not clear which way best fits HAN. As a result, it requires several experiments to identify the best way to extract embeddings. More details are presented in section 5.3.

CHAPTER 2

BACKGROUND

This chapter gives some background information regarding deep learning architectures and Question Answering (QA). For deep learning, we give an overview of commonly used models. The models are Global Vectors (Glove), Gated Recurrent Units (GRUs), and Bidirectional Encoder Representation from Transformer (BERT). In addition, we discuss Hierarchical Attention Network (HAN). Regarding QA, we discuss how QA was being solved before the advancement in deep learning. Then, we consider Knowledge Base (KB) QA systems. After that, we discuss document based question answering. Finally, we provide some discussion on visual QA.

2.1 Deep Learning Architectures

2.1.1 Global Vectors

Glove is developed by Pennington et al. [6]. Glove is used in Natural Language Processing as an embeddings layer. Embeddings refer to mapping words into vector space such that vectors preserve meanings of words [6]. Pennington et al. construct a

word co-occurrence matrix to learn embeddings of words. An entry X_{ij} in the matrix holds the number of times a word_j appears in the context of a word_i. The context is a window of size ten, and it considers five words before the word_i and five words after. This simple matrix is the basis for Glove. Pennington et al. trained Glove on a dataset of six billions tokens. They called the model Global Vectors because it is able to learn the global statistics of the training data.

2.1.2 Gated Recurrent Units

GRUs are type of neural networks that can work with natural languages, and they are developed by Cho at al. [8]. They process a word at a time. GRUs have gating mechanism which allows them to learn long-term dependencies [8]. Instead of encoding every word, they just encode the words that contribute to solving the task being addressed, for example, QA. As illustrated in section 3.2, researchers have been using Bi-directional GRUs (BiGRUs).

BiGRUs consist of two GRUs. One GRU processes input from left to right, and the other GRU processes input from right to left. Then, the outputs of the two models are concatenated. The concatenated vector holds information of left to right and right to left encoding. Hence, it is named bi-directional.

2.1.3 Bidirectional Encoder Representation from Transformer

BERT is proposed by Devlin et al. [7]. BERT consists of 12 layers. Each layer performs attention text encoding [9] followed by a fully connected network. BERT is trained on BookCorpus data having 800 million words [10] and English Wikipedia data having 2,500 million words [7]. BERT has 110 millions parameters, and the large number of parameters allows it to perform well on different NLP tasks [7]. The main component of BERT is the attention based text encoding. Unlike GRUs, attention text encoding processes input text in parallel, so it is faster than GRUs [9].

2.2 Hierarchical Attention Network

HAN is develop by Yang et al. [1]. The goal of the authors was to develop a model for sentiment estimation and topic classification. Figure 2.1 shows the developed HAN. It has two layers of text encoding. After each layer, attention is applied. Attention operation is a dot product followed by a soft-max operation. In the output layer, attention scores are added, and predictions are provided. The main advantage of HAN is the attention operations which enable it to focus on relevant parts of the input. Attention is illustrated in detail in chapter 4 and appendix A. Alpay et al. [2] proposed a HAN model for cloze-style QA.

The proposed model is called HAN pointer sum attention (HAN-ptr). HAN-ptr is shown in figure 2.2. It also has two layers of text encoding and attention operations

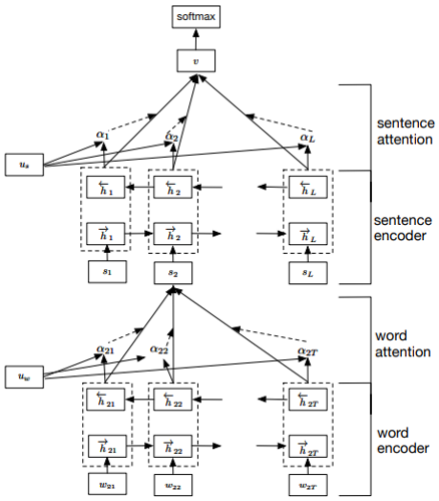

Figure 2.1: HAN for sentiment estimation and topic classification [1].

after each layer. In the output layer, attention scores are added. Attention scores are given to each word in the input context. The word with the highest score is predicted to be the answer. More details about the operations of HAN are presented in chapter 4 and appendix A.

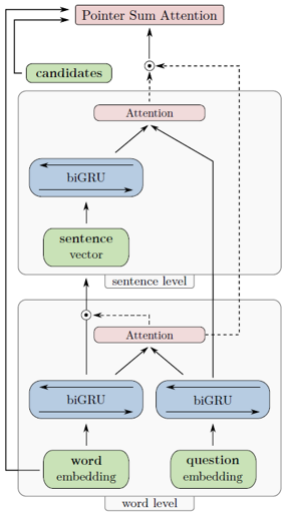

Figure 2.2: HAN for cloze-style QA [2].

2.3 Question Answering

2.3.1 Question Answering before Deep Learning

QA and Natural Language Processing (NLP) in general were harder before the advancement in deep learning. Researchers and engineers had to design complex rules and regular expressions to extract useful information from input text [11], [12]. Moreover, they needed to have several modules working together to answer a single question such as processing questions and retrieving information from Knowledge Bases (KBes) [13], [14], [15]. To design a system, one needed experts not only in NLP but also in the domain targeted by the system, and the systems worked with KBes.

Figure 2.3 shows a block diagram of a typical QA system. Complex rules are applied on input questions to extract information. Then, a KB is searched to retrieve answers. The answers are analyzed to select the best one(s). Finally, the best answer is delivered.

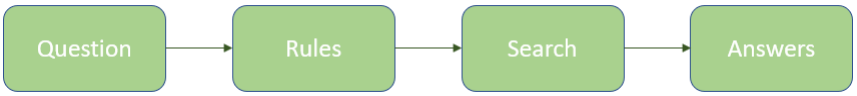

Figure 2.3: A typical QA system before the advancements in deep learning based models.

2.3.2 Knowledge Base Question Answering

KB QA systems answer questions based on some databases (i.e., KBes) holding information regarding the domain addressed by the systems. Information is stored

in specific format in KBes, and the formats are not universal for all KBes [16]. So, KB QA systems need some processing to extract information from input and match it with the KB [17], [18]. After extracting the information, some deep learning techniques (e.g., Recurrent Neural Networks) could be used to select the best piece of information that answers the input question [19]. Although the systems provide good results [20], they only work with the KBes that they designed for.

Figure 2.4 shows a block diagram of a typical KB QA system. Some rules are applied on input questions to extract information. Then, the system searches a KB to retrieve answers. The answers are analyzed using deep learning to select the best one(s). Finally, the system gives the best answer.

Figure 2.4: A typical KB QA system.

2.3.3 Document Based Question Answering

Document based QAs are the simplest QA systems. They do not have KBes to work with, and they do not require complex NLP rules to extract useful information as this task is left for advanced deep leaning models [21]. A document based QA system has two inputs which are a document and a question [22]. Input document and question can be directly processed by recurrent, convolutional, or attention networks [23], [24]. System designers can also have simple NLP processing before the inputs are passed to the networks [25], [26]. Although document based QAs are

simple, they provide good results [27]. However, one drawback is that the systems need large scale data sets and, hence, long training times.

Figure 2.5 shows a block diagram of a typical document based QA system. Optional rules could be applied on input documents and questions to extract some useful information. Then, inputs are processed by deep learning models. This allows the system to give the best answer as it learns interactions between a document and a question. For the optional rules, one might use Part of Speech tags before passing inputs to deep learning models [4].

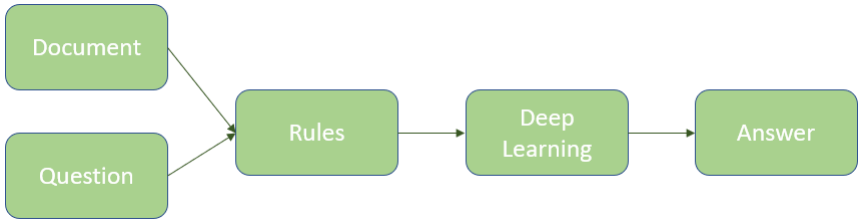

Figure 2.5: A typical document based QA system.

2.3.4 Visual Question Answering

Visual QA (VQA) is an interesting QA application as it involves images and text, and it needs to learn interactions between them. VQA systems accept two inputs, namely, an image and a question [28]. As a result, these systems typically have two sub-modules. The first one considers processing images, and it consists of Convolutional Neural Networks [29]. The second sub-module works with textual contents, and it includes Recurrent Neural Networks [30]. To learn interactions between visual and textual contents, attention is applied on the output of the two

sub-modules [31], [32].

Figure 2.6 shows a block diagram of a typical VQA system. Images and questions are processed independently by two separate sub-modules. The results of the sub-modules are analyzed further by some deep learning model. This gives a VQA system the ability to learn how visual and textual contents interact with each other and, hence, answer questions.

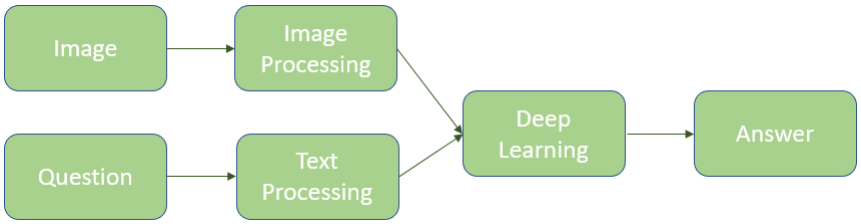

Figure 2.6: A typical VQA system.

CHAPTER 3

LITERATURE REVIEW

Three common cloze-style question datasets will be discussed in this chapter. Also, some related work in cloze-style QA systems and some general approaches towards QA tasks will be briefly reviewed. For each study, we report proposed approach, training data, and training and testing results. For studies addressing cloze-style QA, we mention which optimization algorithm was used to update trainable parameters of proposed models. Inference times of models addressing cloze-style QA were not reported by researchers. In this review, we do not consider large language models such as BERT [7] and Generative Pre-trained Transformer (GPT) [33] as these models are not designed specifically for cloze-style QA. They are meant to be general purpose language models (i.e., could be used in a variety of NLP tasks).

3.1 Cloze-Style Questions Datasets

The three commonly used Cloze-style data are Children's Book Test (CBT), Cable News Network (CNN), and Daily Mail datasets. Recently they have been utilized to

train and validate deep learning based models to answer Cloze-style questions. The following discusses some information about these datasets.

CBT was collected from children books, and Hill et al. [3] argued that children books provide a clear narrative structure. Each example in the data has a 20-sentence context, a 1-sentence query with a missing word, and 10 candidate answers including the true answer. Missing words in query sentences could be one of two types which are named entities and common nouns. So, CBT has two instances, namely, Named Entity (CBT-NE) and Common Nouns (CBT-CN) where the first has 108,719 training, 2,000 validation, and 2,500 test examples; and the second has 120,769 training, 2,000 validation, and 2,500 test examples.

The following is a data point from CBT-CN data. The context is

1 ‘ What is it ? ’

2 answered he .

3 ‘ The ogre is coming after us .

4 I saw him . ’

5 ‘ But where is he ?

6 I do n’t see him . ’

7 ‘ Over there .

8 He only looks about as tall as a needle . ’

9 Then they both began to run as fast as they could , while the ogre and his dog kept drawing always nearer .

10 A few more steps , and he would have been by their side , when Dschemila threw the darning needle behind her .

11 In a moment it became an iron mountain between them and their enemy .

12 ‘ We will break it down , my dog and I , ’ cried the ogre in a rage , and they dashed at the mountain till they had forced a path through , and came ever nearer and nearer .

13 ‘ Cousin ! ’

14 said Dschemila suddenly .

15 ‘ What is it ? ’

16 ‘ The ogre is coming after us with his dog . ’

17 ‘ You go on in front then , ’ answered he ; and they both ran on as fast as they could , while the ogre and the dog drew always nearer and nearer .

18 ‘ They are close upon us ! ’

19 cried the maiden , glancing behind , ‘ you must throw the pin . ’

20 So Dschemil took the pin from his cloak and threw it behind him , and a dense thicket of thorns sprang up round them , which the ogre and his dog could not pass through . ’.

The query line is

21 I will get through it somehow , if I burrow underground , ’ cried he , and very soon he and the XXXXX were on the other side . dog Cousin/cloak/dog/maiden/mountain/needle/path/pin/side/steps

which includes the query, the true answer (i.e., dog), and ten possible answers.

The other two datasets, CNN and Daily Mail, were collected by Hermann et al. [4] from Cable News Network and Daily Mail websites. The authors noticed that articles in CNN and Daily Mail websites are followed by bullet point summaries. So, by removing an entity from one of the bullet points and having the other points as queries, the researchers could construct a document-query-answer triple data. CNN data consists of 380,298 training examples, 3,924 validation examples, and 3,198 testing examples; and Daily Mail data 879,450 training examples, 64,835 validation examples, and 53,182 testing examples.

The following is an example from Daily Mail data. The context is

the ent381 producer allegedly struck by ent212 will not press charges against the " ent153 " host , his lawyer said friday . ent212 , who hosted one of the most - watched television shows in the world , was dropped by the ent381 wednesday after an internal investigation by the ent180 broadcaster found he had subjected producer ent193 " to an unprovoked physical and verbal attack . ".

The query is

producer X will not press charges against ent212 , his lawyer says ..

The answer is

ent193,

and it should replace X in the query. The example includes the words *ent* followed by numbers. These words replace names as names do not contribute to the meaning of the sentences [4].

Three common cloze-style data are CBT, CNN, and Daily Mail datasets. They have been utilized recently to train and validate deep learning based models to answer cloze-style questions. Table 3.1 provides a summary of these datasets.

Table 3.1: Summary of cloze-style question datasets.

Data	***Training Examples***	***Validation Examples***	***Testing Examples***	***Context Length***	***Sentence Length***
CBT-NE	108,719	2,000	2,500	20 sentences	variable
CBT-CN	120,769	2,000	2,500	20 sentences	variable
CNN	380,298	3,924	3,198	variable	variable
Daily Mail	879,450	64,835	53,182	variable	variable

3.2 Cloze-Style Approaches

HAN was developed by Yang et al. [1] with a goal to perform text classification. Alpay et al. [2] proposed two cloze-style QA models based on HAN architecture that utilize Bi-directional Gated Recurrent Units (BiGRUs) to encode input sequences. The main difference between the two models was in the final layer where the first model used a soft-max based classifier to generate scores of predicted classes and the second model used pointer sum attention to add attention scores of words. In this work, the authors utilized Adam optimizer [34]. They used Glove embeddings [6] to converted the input text into vector spaces and CBT data to validate their models. Accuracy scores of the first model on validation and test sets of CBT-NE data were 62.9% and 57.7% respectively, and its scores on validation and test sets of CBT-CN data were 60.4% and 56.4% respectively. Accuracy scores of the second model on validation and test sets of CBT-NE data were 75.5% and 69.9% respectively, and its

scores on validation and test sets of CBT-CN data were 69.1% and 67.7% respectively.

Fu et al. [35] proposed a model to answer cloze-style questions. The model was based on BiGRU and intra-attention. Intra-attention was computed by applying dot product between a word and every other word in a context. Intra-attention was used to model long-term dependencies. The output (e.g., an answer) was provided via inter-attention mechanism. The model was optimized using Adam. The authors utilized CBT data to validate their model. The model achieved accuracy scores of 77.7% and 74.2% on validation and test sets of CBT-NE data respectively, and 75.9% and 74.5% on validation and test sets of CBT-CN data respectively.

Fu and Zhang [36] developed a model that addresses cloze-style questions. The model utilized multiple latent semantic spaces to process queries and documents. Also, the model utilized bit-level attention instead of token-level attention. The bit-level attention was computed using a fully connected layer with output dimensionality equals to the size of the embeddings. So, the attention produced a score to every number in an embeddings vector. Fu and Zhang argued that bit-level attention is more effective than token-level attention. To train their model, Fu and Zhang used Adam optimizer. They validated their model using CBT data. The model achieved accuracies of 78.1% on validation and 73.1% on test sets of CBT-NE and 73.9% on validation and 71.5% on test sets of CBT-CN.

Dhingra et al. [37] proposed a Gated Attention (GA) and BiGRU based model

for cloze-style question answering. Gates in attention mechanism instructed the attention whether to apply summation, concatenation, or multiplication of the output provided by BiGRUs. The model had four layers of BiGRUs and GA. Dhingra et al. used Adam optimizer to train the model. The researchers validated their model on CBT-NE and CBT-CN. The model achieved accuracy scores of 78.5% and 74.9% on validation and test sets of CBT-NE data and 74.4% and 70.7% on validation and test sets of CBT-CN data.

Kadlec et al. [38] proposed an ensemble model based on attention mechanism and BiGRU to provide answers for cloze-style questions. The model was validated on CBT data, and its parameters were updated using Adam optimizer. The model achieved 76.2% and 71% accuracy scores on validation and test sets of CBT-NE data, and 72.4% and 67.5% accuracy scores on validation and test sets of CBT-CN data.

Hermann et al. [4] proposed an attention mechanism and bidirectional long short-term memory (BiLSTM) based model for cloze-style questions. Their model applied attention on context and query after they were fully encoded. For parameter updates, Hermann et al. utilized Root Mean Square Propagation (RmsProp) algorithm [39]. The model was validated on data from CNN and Daily Mail websites collected by the authors. The developed model achieved accuracy scores of 61.6% and 63% on validation and test sets of CNN data, and 70.5% and 69% on validation and test sets of Daily Mail data.

Shen et al. [5] proposed an attention mechanism and BiLSTM based model to address cloze-style questions. The authors introduced some reinforcement learning techniques to dynamically instruct the model when to stop reading a document. Adam optimizer was used to guide training process. The model was validated on CNN and Daily Mail data. Proposed model got accuracies of 72.9% and 74.7% on CNN validation and test sets, and 77.6% and 76.6% on Daily Mail validation and test sets.

The presented work shows that there are several deep learning approaches towards Cloze-style question answering systems. However, only one model—proposed by Alpay et al. [2]—was based on HANs. Table 3.2 summarizes presented literature.

Table 3.2: Summary of approaches towards cloze-style question answering. AM stands for attention mechanism, and DM stands for Daily Mail

References	**Architectures**	**Optimizers**	**Data**	**Accuracy Scores**	
				Validation	**Test**
Alpay et al. [2]	HAN and BiGRU	Adam	CBT-NE	75.5%	69.9%
			CBT-CN	69.1%	67.7%
Fu et al. [35]	BiGRU and intra-attention	Adam	CBT-NE	77.7%	74.2%
			CBT-CN	75.9%	74.5%
Fu and Zhang [36]	Bit-level attention	Adam	CBT-NE	78.1%	73.1%
			CBT-CN	73.9%	71.5%
Dhingra et al. [37]	Gated Attention and BiGRU	Adam	CBT-NE	78.5%	74.9%
			CBT-CN	74.4%	70.7%
Kadlec et al. [38]	AM and BiGRU	Adam	CBT-NE	76.2%	71.0%
			CBT-CN	72.2%	67.5%
Hermann et al. [4]	AM and BiLSTM	RmsProp	CNN	61.6%	63.0%
			DM	70.5%	69.0%
Shen et al. [5]	AM and BiLSTM	Adam	CNN	72.9%	74.7%
			DM	77.6%	76.6%

3.3 General Approaches

There are several approaches that have been used to design QA systems. The following discusses some of recent work that is related to this research.

Du et al. [40] proposed a biomedical QA system consisting of four layers. The first layer contains pre-trained BERT [7], the second layer Bidirectional Long-Short Term Memory (BiLSTM), and the third layer an attention mechanism. The fourth layer is the output layer. Since BERT was pre-trained, the authors fine-tuned it with the BioASQ [41] dataset to improve its performance. The dataset had 1,799 questions related to the biomedical field. The model was then tested using a biomedical related dataset containing 1,799 questions and achieved an accuracy score of 33% and a Mean Reciprocal Rank (MRR) score of 38%.

Li et al. [42] developed a Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) based encoders QA system. The model was trained on the NLPCC2016 which is a chinese language dataset containing training and testing sets [43]. The training set has 181,882 question and answer pairs. The testing set has 122,531 question and answer pairs. The model achieved a Mean Average Precision (MAP) score of 84.4%, an MRR score of 84.5%, and a precision of 77.1%.

Xiao et al. [44] proposed a QA system that utilizes both BiLSTMs and attention mechanism. The input text was converted into vector space using Glove word

embeddings [6]. Some features were added to the encoded paragraphs such as Part of Speech (POS) and named entity tags. The authors used Stanford Question Answering Dataset (SQuAD) [45] to train their model. The model achieved an F1 score of 79.7% and an Exact Match (EM) score of 71.4% on that dataset.

Peng and Liu [46] proposed an attention-based CNN model for QA tasks. The main goal of the model is to re-rank candidate answers and remove factoid answers. This task was accomplished by obtaining word-level and phrase-level interactive. The model then generates probabilities to re-rank candidate answers. The model was trained on TrecQA [47] which has 4,718 training examples, 1,148 validation examples, and 1,517 testing examples. The model achieved an MAP score of 68.8% and an MMR score of 77.3% on this dataset.

Yang et al. [48] designed an attention based Neural Matching Model (aNMM) to tackle QA. The model mapped a question and an answer to their embedding representations. Then, it constructed a matching matrix with rows and columns equal to the lengths of the question and answer respectively. Element e_{ij} in the matrix measured how similar the i^{th} word in the question to the j^{th} word in the answer. The similarity was computed using cosine similarity. After that, the model applied attention and feed forward network. The final output of the model was a score between 0 and 1 that indicated how likely the provided answer would be the right answer. The model was trained and tested on TrecQA dataset. The model could achieve MAR and MMR scores of 73.85% and 79.95% respectively.

Bao et al. [49] developed a Knowledge Base (KB) based system to address QA. The system translated questions to meaning representations using Cocke–Younger–Kasami (CYK) parsing algorithm. This step was important as it allowed the system to retrieve answers from the KB, and it was the main contribution of the research. The authors used WebQuestions data [50] to validate their system. The system was able to produce an accuracy score of 37.5%.

As discussed above, there are different approaches to build deep learning based systems for QA tasks. However, attention mechanism is the common approach. Table 3.3 summarizes presented literature.

Table 3.3: Summary of general approaches towards QA.

References	***Architectures***	***Data***	**Accuracy**	
			Metrics	**Scores**
Du et al. [40]	BERT and BiLSTM	BioASQ	accuracy, MRR	33%, 38%
Li et al. [42]	RNN and CNN	NLPCC2016	MAP, MRR	84.4%, 84.5%
Xiao et al. [44]	attention mechanism and BiLSTM	SQuAD	EM, F1	71.4%, 79.7%
Peng and Liu [46]	attention mechanism and CNN	TrecQA	MAP, MMR	68.8%, 77.3%
Yang et al. [48]	attention mechanism and matching network	TrecQA	MAP, MMR	73.85%, 79.95%
Bao et al. [49]	Knowledge Base	WebQuestions	accuracy	37.5%

CHAPTER 4

PROPOSED MODEL

We describe proposed model in some details. We discuss overall operations of proposed model. Then, we illustrate first and second level attentions. We do not provide mathematical details in this chapter. Instead, we present them in appendix A.

4.1 Model Description

Our proposed model is based on HAN pointer sum attention (HAN-ptr) model for cloze-style developed by Alpay et al.[2]. Figure 4.1 shows a block diagram of proposed model. Colored boxes indicate inputs or outputs, and white boxes indicate operations. Dashed arrow going from a BiGRU indicates all hidden states of that Bi-GRU, and solid arrow going from a BiGRU indicates last hidden state of that BiGRU.

First, context sentences and query are encoded word by word using two Bi-GRUs. Then, attention is performed on all hidden states of encoded sentences

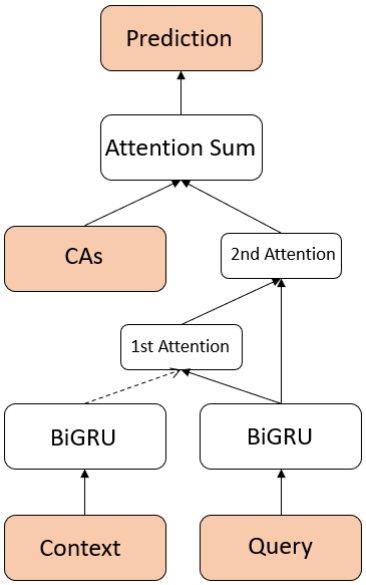

Figure 4.1: A block diagram of proposed model where CAs stands for Candidate Answers.

and last hidden state of encoded query. This results in an attended context vector (context after attention is performed on it) that has relevant information to query. Another attention is performed on the encoded query and attended context. This enhances the attended context vector because the operation reduces values of irrelevant information compared to relevant. Then, first and second attention scores are multiplied which results in word scores. After that, attention sum [38] is performed on candidate answers (e.g., a list of 10 possible answers) and word scores.

Attention sum just adds scores of a word. To illustrate, a context could have repeated words, and scores of these words might be different depending on their locations in the context. As an example, if a context has the word ”school” three times with scores 0.05, 0.1, and 0.07, then the score of ”school” would be 0.22. The output of HAN is the word with maximum score, and it is predicted to be the answer to query.

4.2 First Attention

First attention is illustrated in figure 4.2. Inputs to the operation are encoded context and encoded query. Encoded context is a matrix with shape $(sequence, output\_space * 2)$ where *sequence* is the number of words in a context, and *output_space* is the output dimension of a GRU unit. It is multiplied by two because we concatenate outputs of two GRUs. Encoded query is a vector with shape $(1, output\_space * 2)$. Dot product is applied between encoded query and every row in encoded context. Dot product returns scalar values, and these values are used as similarity metrics between encoded query and each word in encoded context. So, the dot product results in a vector of size $(sequence, 1)$, and a soft-max operation is applied on this vector. Soft-max operation produces attention scores of words. For example, row_i in attention scores vector has the score of i^{th} word in input context. Finally, each row in encoded context is multiplied by the corresponding score using element-wise multiplication which gives attended context. As indicated by colored boxes in figure 4.2, outputs of first attention are attention scores and attended context.

4.3 Second Attention

Operations of second attention are similar to first attention and are shown in figure 4.3. Inputs are attended context, encoded query, and attention scores (input attention scores). Dot product is applied between encoded query and every row in

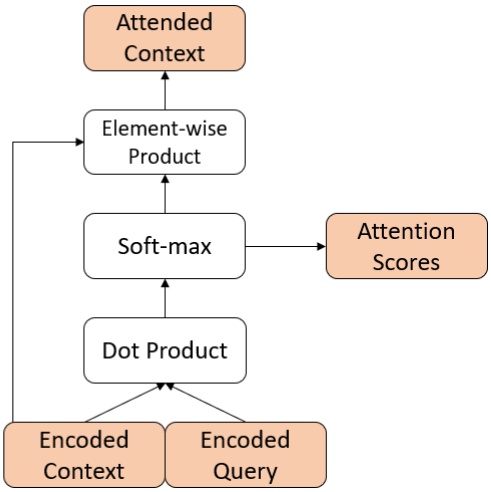

Figure 4.2: A block diagram of first attention.

attended context. The result is a vector of size $(words, 1)$, and a soft-max operation is applied on this vector to get attention scores (obtained attention scores). After that, element-wise multiplication is performed between obtained attention scores and input attention scores. Although the operation reduces values of input attention scores, it decrease values of irrelevant words much more since values of these words will be multiplied by smaller values. The output of second attention is the result of element-wise multiplication which is word scores vector.

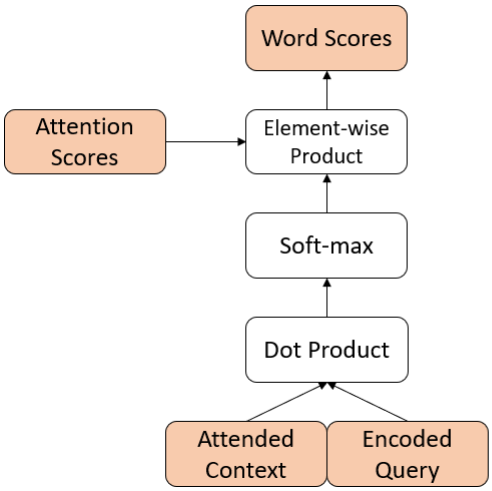

Figure 4.3: A block diagram of second attention.

CHAPTER 5

EXPERIMENTAL SETUP

In this chapter, we describe data processing and design of experiments. We have two sets of experiments. The first set is designed to compare one-layer text encoding HAN models and two-layer text encoding HAN models. The comparison is based on inference times and accuracy scores. The second set is designed to measure effects of different BERT embeddings on HAN models' accuracy scores. Furthermore, we discuss some details regarding model implementation and training environment.

5.1 Data Processing

In this section, we describe utilized data. This includes structure and pre-processing of data. We utilize CBT, CNN, and Daily Mail datasets.

5.1.1 CBT

We are considering both CBT-NE and CBT-CN. Each example of datasets has 21 lines. The first 20 lines are context sentences, the 21^{st} line has three parts which

are a query, true answer, and a list of 10 candidate answers.

CBT data is well organised. So, pre-processing is minimal. For each dataset, we process data line by line as the following. For each line, we remove line numbers and new line character. After that, we store each word of the line in a hash data structure. Each entry in the structure is a tuple consisting of a word and its frequency. The frequency of a word is updated as the processing progresses. For example, if a word in some line has a frequency of 2, then its entry in the hash data structure would be 2. If the same word appears in another line with a frequency of 3, then its frequency would be updated to be five (e.g., 2 + 3). At the end of this process, the hash has an entry for each word in a dataset (e.g., CBT-NE or CBT-CN) and its corresponding frequency.

We utilize these frequencies to sort words from the most common to the least common. This allows us to assign Identification (ID) numbers to words. IDs range from 0 to (max - 1) where max is the total number of words in a dataset. ID of 0 is assigned to the most common word in a dataset, and ID of (max - 1) is assigned to the least common word in the dataset. Then, we convert data files from text to integers by replacing each word by its ID. This process is not mandatory, but it results in smaller file sizes. For instance, the original size of CBT-NE training data is 237 Megabyte (MB), while the size of the corresponding ID file is 168 MB.

After we read ID files, we convert them into vector space using an embedding

matrix. Originally, embedding matrix is indexed by words row-wise. For instance, row_i has the vector representation of word_i. So, we replace word indices by the corresponding IDs.

5.1.2 CNN and Daily Mail

CNN and Daily Mail data have complex structure compared to CBT. For instance, they have variable length documents, and a document might have several questions. So, we transform CNN and Daily Mail data so that they would have the same structure as CBT. To illustrate, each example of transformed CNN and Daily Mail data has 21 lines. Context sentences are the first 20 lines, the 21^{st} line contains a question, true answer, and a list of 10 possible answers.

The process of transforming data is as the following. We ignore all questions and answers and only consider documents. From the total documents of a dataset (e.g., CNN or Daily Mail), we extract 2000 documents as validation documents and 2500 documents as test documents. For each document, we extract the first 20 lines so that we would have a 20-line context. For the 21^{st} line (query line), we extract a word as an answer to a query, and the query is the same line after extracting the word. After that, we add a list of 10 possible answers to the query line. The list consists of the true answer and nine words randomly sampled from the context. As a result of this process, each document would have the same structure as CBT data. Then, we convert words to IDs as described in section 5.1.1.

We consider some criteria to transform data. If a document has more than 21 lines, the extra lines are ignored. If a document has less than 21 lines, the document is ignored. To extract a word from the query line, the word must be a noun or a verb. To ensure this requirement, we utilize a Part of Speech (POS) tagger. If a query line does not have a noun or a verb, the corresponding document is ignored. These requirements reduce sizes of training, validation, and test sets. Resulted samples for CNN are 61640 training samples, 1388 validation samples, and 1758 test samples. Training, validation, and test sets of Daily Mail have 86612, 1738, 2191 samples respectively.

5.2 Text Encoding Layers

For the first set of experiments, we develop two HAN models. First model is our proposed model. Second model is a replication of model developed by Alpay et al. [2]. We develop it to the best of our knowledge and according to the description provided by Alpay et al. Replicated model serves as a baseline for proposed model. The only difference between proposed and baseline models is the number of text encoding layers. For instance, proposed model has one layer of text encoding. On the other hand, baseline model has two layers. Consequently, observed inference times and accuracy scores would be caused by the difference in number of text encoding layers. Experiments are done using the four datasets discussed in section 5.1. Table 5.1 illustrates objects, independent variables, and dependent variables of experiments.

Table 5.1: Illustration of text encoding layers' experiments.

Objects	**Variables**	
	Independent	***Dependent***
HAN models	Number of text encoding layers	Inference times and accuracy scores

5.3 BERT Embeddings

We experiment with embeddings extracted from pre-trained BERT model [7]. BERT was trained on BookCorpus data having 800 million words [10] and English Wikipedia data having 2,500 million words [7]. BERT has 12 layers, and each layer has an output dimensionality of 768. BERT is designed to take two sentences as input and to generate vector representations (embeddings) of them. There are different ways to extract embeddings from BERT's layers.

Devlin et al. [7] found that taking concatenation of outputs of last four layers resulted in the best performance in Named Entity Recognition task. We try this approach, but it does not provide us with accuracy scores of more than 70% as shown in table 6.3. Consequently, we take outputs of first layer, average of sixth and seventh (middle layer), and last layer. We consider these layers because we want to observe how embeddings extracted from deeper layers affect HAN's accuracy. To obtain embeddings for all four datasets, we extract unique words from datasets and pass these words individually to BERT so that BERT generates embeddings for words instead of sentences. Table 5.2 illustrates objects, independent variables, and dependent variables of experiments.

Table 5.2: Illustration of BERT embeddings experiments.

Objects	**Variables**	
	Independent	***Dependent***
HAN models	Embeddings layers of BERT	Accuracy scores

5.4 Hyper-Parameters and Training

We set hyper-parameters as the following. We have GRUs with output spaces of 384 and Glove embeddings with 200 dimensions. These values are among the best values experimented by Alpay et al. [2]. We experiment with output spaces and embeddings dimensions for one-layer text encoding HAN to confirm our choices. Experiments' results are presented in appendix C.

Regarding model optimizer, we use Adam [34] optimizer with learning-rate of 0.001, β_1 of 0.9, and β_2 of 0.999. We set these parameters after experimenting with different values. Results of experiments are presented in appendix C. Gradient clipping value of 10 is used. Training is done in batches each of size 64 samples. Models are evaluated every 200 batches. Training process is terminated if a model starts to overfits its training data.

We develop HAN models using TensorFlow. Models are trained on Colaboratory (Colab). Colab is a cloud-based Python environment. Colab provides a machine with a Central Processing Unit (CPU) of type Xeon with a 2.3-Gigahertz (GHz) clock frequency and a cache memory of size 45 Megabytes (MB), a Random Access Memory (RAM) of approximately 25.5 Gigabytes (GB), an NVIDIA Graphics Processing

Unit (GPU) of type T4 with a 16-GB RAM, and a disk size of approximately 68.4 GB.

CHAPTER 6

RESULTS

This chapter presents results of both sets of experiments. For text encoding layers, we evaluate proposed and baseline models on test sets of all four datasets. To assess our results, we compare inference times and accuracy scores of proposed model against two models, namely, Gated Attention Readers by Dhingra et al. [37] and Attention Sum Reader by Kadlec et al. [38]. Regarding BERT embeddings, we experiment with embeddings extracted from different layers using CBT-NE data. Then, we take the best embeddings to train models on the other datasets. This step is important to ensure that observed behaviour could generalize to other datasets.

6.1 Test Results of Text Encoding Layers

Tests are performed by evaluating proposed and baseline models on batches of 64 samples from test data. We consider two criteria which are inference times and accuracy scores. Inference time is expressed in seconds, and it is the time needed by a model to process and answer a single test sample. Accuracy is computed as the total

number of correctly identified answers divided by the total number of predictions in a batch. CBT-NE and CBT-CN both have 2500 test samples. On the other hand, CNN has 1758 test examples, and Daily Mail has 2191 test examples. Consequently, models are tested on 39 batches (i.e., floor(2500 / 64)) of CBT-NE and CBT-CN test sets, 27 batches of CNN test set, and 34 batches of Daily Mail test set.

Table 6.1 shows test results of the models. For CBT-NE test data, proposed model gets an average inference time of 0.01255 seconds and an average accuracy score of 70.47%. Baseline model gets an average inference time of 0.02129 seconds and an average accuracy score of 68.99%. This corresponds to inference time reduction of 41.05% and accuracy increase of 2.14%. Table 6.1 indicates that proposed model has an average inference time reduction of 41.39% and an average accuracy increase of 0.4142%.

6.2 Additional Evaluation

To have a more comprehensive evaluation, we replicate two state-of-the-art models which are Gated Attention Readers and Attention Sum Reader models developed by Dhingra et al. [37] and Kadlec et al. [38], respectively, and compare their inference times and accuracy scores with our model's. The results are illustrated in table 6.2. Accuracy scores of all models are similar. However, inference times of proposed model are better than Gated Attention Readers'. On the other hand, proposed model and Attention Sum Reader have close inference times.

Table 6.1: Test results of proposed and baseline models. IT stands for inference time, and it is measured in seconds. AS stands for accuracy score.

Test Data	**Models**	**Results**	
		Average IT	**Average AS**
CBT-NE	Proposed model	0.01255	70.47%
	Baseline model	0.02129	68.99%
	Reduction in IT	41.05%	-
	Increase in AS	-	2.14%
CBT-CN	Proposed model	0.01268	67.42%
	Baseline model	0.02138	67.54%
	Reduction in IT	40.69%	-
	Increase in AS	-	-0.1779%
CNN	Proposed model	0.01738	96.87%
	Baseline model	0.03039	97.01%
	Reduction in IT	42.8%	-
	Increase in AS	-	-0.1466%
Daily Mail	Proposed model	0.0118	90.34%
	Baseline model	0.02002	90.48%
	Reduction in IT	41.05%	-
	Increase in AS	-	-0.1569%
Average results	Reduction in IT	41.39%	-
	Increase in AS	-	0.4142%

It should be noted that accuracy scores of the replicated models in table 6.2 are different from the results of the same models in table 3.2. This is because we only consider the developed architectural designs (i.e., the main contributions of both studies) of the models and ignore additional enhancements such as character level embeddings and ensemble of models. Also, our implementations of the models use larger embeddings and BiGRU units. We follow this approach because we want our model and the replicated models to differ only on the architectural designs so that observed scores are not caused by other factors.

Table 6.2: Test results of proposed model, Gated Attention Readers, and Attention Sum Reader. IT stands for inference time, and it is measured in seconds. AS stands for accuracy score.

Test Data	**Models**	**Results**	
		Average IT	**Average AS**
CBT-NE	Proposed model	0.01255	70.47%
	Gated Attention Readers	0.03323	70.75%
	Attention Sum Reader	0.0142	69.76%
CBT-CN	Proposed model	0.01268	67.42%
	Gated Attention Readers	0.03294	67.42%
	Attention Sum Reader	0.0144	64.78%

6.3 Test Results of BERT embeddings

Table 6.3 shows validation and test results of experimenting with embeddings taken from different layers. We utilize CBT-NE data for training and testing. The best performing embeddings are extracted from BERT's first layer. In addition, table 6.4 presents results of training HAN models on all four datasets using embeddings extracted from BERT's first layer.

Table 6.3: Results of experimenting with different BERT's embeddings on CBT-NE data.

Layers	**Results**	
	Validation Set	**Test Set**
First	74.85%	69.07%
Mid	70.82%	64.26%
Last	69.35%	62.58%
Last four layers	68.08%	63.95%

Table 6.4: Results of experimenting with BERT's first layer embeddings on all datasets.

Data	**Results**	
	Validation Set	**Test Set**
CBT-NE	74.85%	69.07%
CBT-CN	69.76%	67.22%
CNN	96.53%	96.61%
DM	89.87%	90.42%

CHAPTER 7

ANALYSIS AND DISCUSSION

In this chapter, we provide interpretation of the obtained results. We test hypotheses corresponding to the first and second research questions. We also discuss results of BERT embeddings and threats to validly of the research. The last section of the chapter presents an issue with training models.

7.1 Hypothesis Testing

We conduct statistical tests on results of models to test hypotheses of this research. We utilize Shapiro–Wilk test [51] to test normality of results. Outcomes of Shapiro–Wilk are presented in table 7.1. Test shows that some of results are not normally distributed. Consequently, we conduct a non-parametric test to test hypotheses, and test is Mann–Whitney U test [52]. For normally distributed results, we use t-test.

Results of statistical tests are shown in table 7.2. All statistical tests performed

Table 7.1: Shapiro–Wilk test results. IT stands for inference time, and AS stands for accuracy score.

Data	Model	Samples	p-value	Normally distributed?
CBT-NE	Proposed model	IT	0.007366	No
		AS	0.0674	Yes
	Baseline model	IT	0.1441	Yes
		AS	$2.19 * 10^{-15}$	No
CBT-CN	Proposed model	IT	0.2515	Yes
		AS	0.3864	Yes
	Baseline model	IT	0.04879	No
		AS	0.5337	Yes
CNN	Proposed model	IT	0.02423	No
		AS	$13.17 * 10^{-4}$	No
	Baseline model	IT	0.02914	No
		AS	0.06955	Yes
Daily Mail	Proposed model	IT	$6.258 * 10^{-8}$	No
		AS	0.0476	No
	Baseline model	IT	0.1433	Yes
		AS	$19.73 * 10^{-4}$	No

on inference times indicate that inference times of proposed model are different from baseline model's. So, H_{01} is rejected, and there is evidence that one-layer text encoding HAN model and two-layer text encoding HAN model have different average inference times. The answer to the first research question is "one-layer text encoding, on average, reduces inference time by 41.39% compared to two-layer text encoding".

Table 7.2: Results of statistical tests. IT stands for inference time, and AS stands for accuracy score.

Data	Samples	Test	p-value	Same distribution?
CBT-NE	IT	Mann–Whitney U	$1.539 * 10^{-14}$	No
	AS	Mann–Whitney U	0.04089	No
CBT-CN	IT	Mann–Whitney U	$1.539 * 10^{-14}$	No
	AS	t-test	0.9274	Yes
CNN	IT	Mann–Whitney U	$1.515 * 10^{-10}$	No
	AS	Mann–Whitney U	0.4708	Yes
Daily Mail	IT	Mann–Whitney U	$1.846 * 10^{-12}$	No
	AS	Mann–Whitney U	0.4228	Yes

On the other hand, statistical tests done on accuracy scores show that accuracy scores of proposed and baseline models are the same. The only exception is results obtained from CBT-NE test data which have a p-value of 0.04089. Consequently, H_{02} is accepted, and there is evidence that one-layer text encoding HAN and two-layer text encoding HAN have the same average accuracy score. The answer to the second research question is "one-layer text encoding HAN and two-layer text encoding HAN result in the same average accuracy score".

7.2 BERT embeddings

According to table 6.3, first layer of BERT gives the best word embeddings. Table 6.4 shows validation and test results of BERT's first layer word embeddings on all four datasets. Results are very close to the ones obtained using Glove embeddings, table 6.1. Consequently, BERT embeddings do not improve HAN's accuracy compared to Glove embeddings.

The reason of such behaviour is that BERT is designed to work with sentences rather than single words. We use BERT with words because it is not feasible to use BERT for sentence embeddings. To illustrate, there are two possibilities to have sentence embeddings with BERT. The first approach is to extract all unique sentences in a dataset, generate embeddings for them, and then store the embeddings on a hash object to be used as a lookup table. This approach has two drawbacks.

First, extracting all unique sentences in a dataset results in taking all sentences in that set, so one would end up storing the whole set in the lookup table. The second issue is that the model cannot work in a real world setting as the lookup table does not include entries for sentences that are not part of the training data.

The other approach is to add BERT as an embeddings layer instead of embeddings lookup tables. This approach was used by Du et al. [40]. We argue that this approach has redundant computations. Devlin et al. [7] showed that BERT could be fine-tuned on a QA task. As a result, BERT could be used as a QA model instead of having it as a sub-model (e.g., embeddings layer) on another QA model. However, BERT is much larger than current QA models. For example, proposed HAN has 2.7 million parameters, while BERT has 110 million parameters. So, it is not feasible to replace current approaches with BERT.

One possible way to advance current approaches is to design attention based text encoding layers. As shown by Vaswani et al. [9], attention text encoding is much faster and provides better results compared to recurrent text encoding. Regarding number of parameters, the two techniques are in the same complexity class [9].

7.3 Threats to Validity

Experiments do not have construct validity because models are sufficiently trained. Experiments do not have internal validity as, for each experiment, only one indepen-

dent variable is changed while other factors are fixed. As a consequence, observed behaviours of models are caused by change in corresponding independent variable. Regarding external validity, experiments have this type of threat since models are trained and tested on two common datasets (i.e., CBT-NE and CBT-CN) and modified versions of CNN and Daily Mail data, so results of experiments cannot be generalized.

7.4 Issue with Training

All trained models encounter local optimal points while training. They reach the points when the loss is around 8. To escape the local optimal points when trained on CBT data, our model, Gated Attention, and attention sum need three hours while baseline model needs four hours. For CNN and Daily Mail data, our model needs six hours, and baseline model needs eight hours. Gated Attention and attention sum are not able to escape the local points even after more than 24 hours of training. For this reason, we do not include their results on table 6.2.

CHAPTER 8

CONCLUSION

QA has gained researchers' attention recently. One reason is the advancement of deep learning architectures. Cloze-style QA is a QA task where a model is fed a context and a query as inputs. The model should infer an answer to the query based on the context. As a candidate model, HAN model could be utilized for such task. We develop a HAN model with a single layer of text encoding. We compare our model against a baseline model having two-layers of text encoding. Results show that developed model has better inference times, while it maintains accuracy scores. This confirms our argument in chapter 1 which states that one layer of text encoding is enough to learn interactions between words as Glove embeddings already capture meanings of individual words.

Moreover, we study effects of different embeddings extracted from different layers of BERT on HAN's performance in terms of accuracy scores. We find that BERT embeddings do not provide better performance over Glove embeddings. This is a result of passing single words to BERT instead of sentences. As future work, we are

going to design attention based text encoding. Attention text encoding is a promising technique which could replace recurrent units. This would improve NLP systems significantly in terms of both inference times and accuracy scores.

APPENDIX A

MATHEMATICAL DETAILS

This section illustrates equations utilized to implement proposed and baseline models. Both models have two layers of attention. First layer of both models have two Bi-directional Recurrent Gated Units (BiGRUs) and an attention operation. These operations are defined mathematically as:

$$encoded\_doc = all\_states(BiGRU(doc)) \tag{A.1}$$

$$encoded\_query = last\_state(BiGRU(query)) \tag{A.2}$$

$$dot\_product = encoded\_query^T \cdot encoded\_doc \tag{A.3}$$

$$attention\_level_{1i} = \frac{exp(dot\_product_i)}{\sum_{j=1}^{sequence} exp(dot\_product_j)} \tag{A.4}$$

$$attended\_doc_i = attention\_level_1 \cdot encoded\_doc_i \tag{A.5}$$

where *doc* stands for document, i in equation A.4 corresponds to i^{th} element of vector $attention\_level_1$, and *sequence* in equation A.4 is the size of vector *dot_product* which equals to the maximum number of words in all documents. It should be noted that equation A.1 returns concatenated hidden states of BiGRU. Consequently, *encoded_doc* is a matrix of dimension $(sequence, output\_space * 2)$ where *output_space* is the output dimension of a GRU unit. Equation A.2 returns concatenated last state of BiGRU. So, *encoded_query* is a vector of dimension $output\_space * 2$. Equation A.3 performs dot product between transpose of *encoded_query* and *encoded_doc* which results in a vector of dimension *sequence*. Equation A.4 performs a soft-max operation on vector *dot_product*. Finally, equation A.5 performs element-wise product between $attention\_level_1$ and i^{th} row of *encoded_doc*. The result of equation A.5 is matrix *attended_doc* with dimension $(sequence, output\_space * 2)$ (e.g., as the dimensionality of *encoded_doc*).

For second layer, proposed model does not have text encoding operation, while baseline model does. So, proposed model proceeds by performing the following set of operations:

$$dot\_product = encoded\_query^T \cdot attended\_doc \tag{A.6}$$

$$attention\_level_{2i} = \frac{exp(dot\_product_i)}{\sum_{j=1}^{sequence} exp(dot\_product_j)} \tag{A.7}$$

$$word\_scores = attention\_level_2 \cdot attention\_level_1 \tag{A.8}$$

$$attention\_sum = \sum_{word \in CA} \sum_{j=1}^{sequence} ((word\_scores_j) and (word == doc_j) \tag{A.9}$$

$$prediction = CA(index\_of\_argmax(attention\_sum)) \tag{A.10}$$

where CA in equation A.9 stands for candidate answers, and it is a list of 10 possible answers. Equation A.6 performs dot product between transpose of *encoded_query* and *attended_doc*. The resulted vector *dot_product* has a dimensionality of *sequence*. Equation A.7 computes second level attention vector $attention\_level_2$ with dimensionality of *sequence*. Equation A.8 computes word scores by performing element-wise product between $attention\_level_2$ and $attention\_level_1$. Vector *word_scores* has a dimension of *sequence*. Equation A.9 performs attention sum operation by summing all scores of a word $\in CA$. Consequently, *attention_sum* has 10 entries. Each entry is a score that the corresponding word in CA is the answer to query. Equation A.10 takes index i of maximum score and predicts CA_i as the answer to query.

Regarding baseline model, it performs the following set of operations:

$$encoded\_doc = all\_states(BiGRU(attended\_doc)) \tag{A.11}$$

$$dot\_product = encoded\_query^T \cdot encoded\_doc \tag{A.12}$$

$$attention\_level_{2i} = \frac{exp(dot\_product_i)}{\sum_{j=1}^{sequence} exp(dot\_product_j)} \tag{A.13}$$

$$word\_scores = attention\_level_2 \cdot attention\_level_1 \tag{A.14}$$

$$attention\_sum = \sum_{word \in CA} \sum_{j=1}^{sequence} ((word\_scores_j) and (word == doc_j) \tag{A.15}$$

$$prediction = CA(index\_of\_argmax(attention\_sum)) \tag{A.16}$$

Equation A.11 performs text encoding, and *encoded_doc* is concatenation of hidden states of BiGRU, and it is a matrix of dimension $(sequence, output\_space * 2)$. Description of equations A.12-A.16 follow exactly the same description of equations A.6-A.10. It should be noted that equation A.6 performs dot product between transpose of *encoded_query* and *attended_doc*, whereas equation A.12 performs dot product between transpose of *encoded_query* and *encoded_doc*.

APPENDIX B

DISTRIBUTION OF TRUE ANSWERS

True answers are not evenly distributed for all datasets. So, datasets are not balanced. For instance, figures B.1a, B.1b, and B.1c show frequencies of true answers of CBT-NE train, validation, and test data. The figures show that answers are not equally distributed. Frequencies of other datasets are shown in figures B.2, B.3, and B.4.

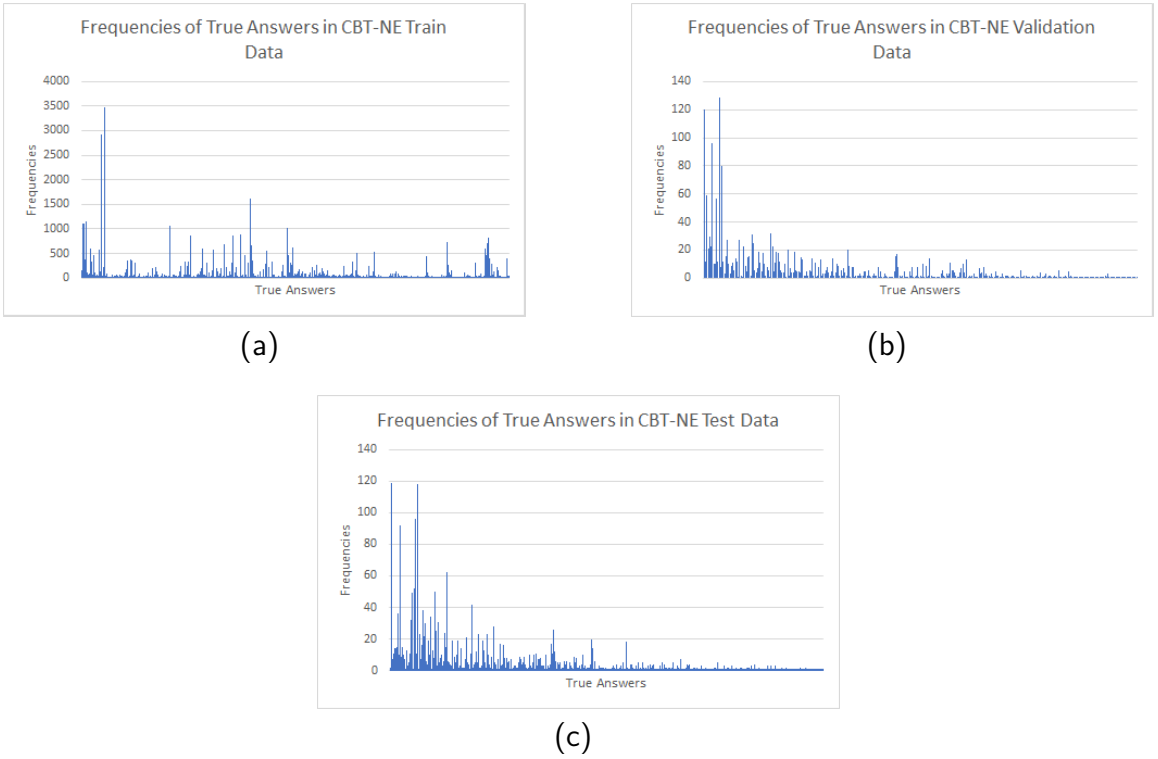

Figure B.1: Frequencies of true answers of CBT-NE (a) train data, (b) validation data, and (c) test data.

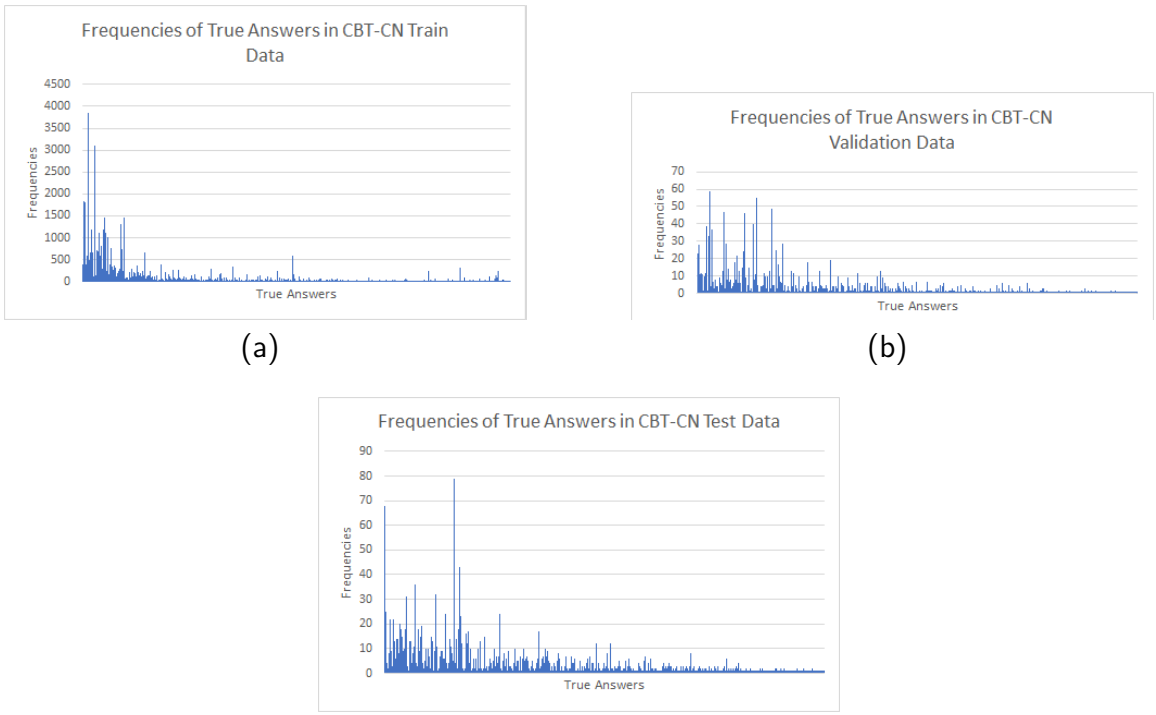

Figure B.2: Frequencies of true answers of CBT-CN (a) train data, (b) validation data, and (c) test data.

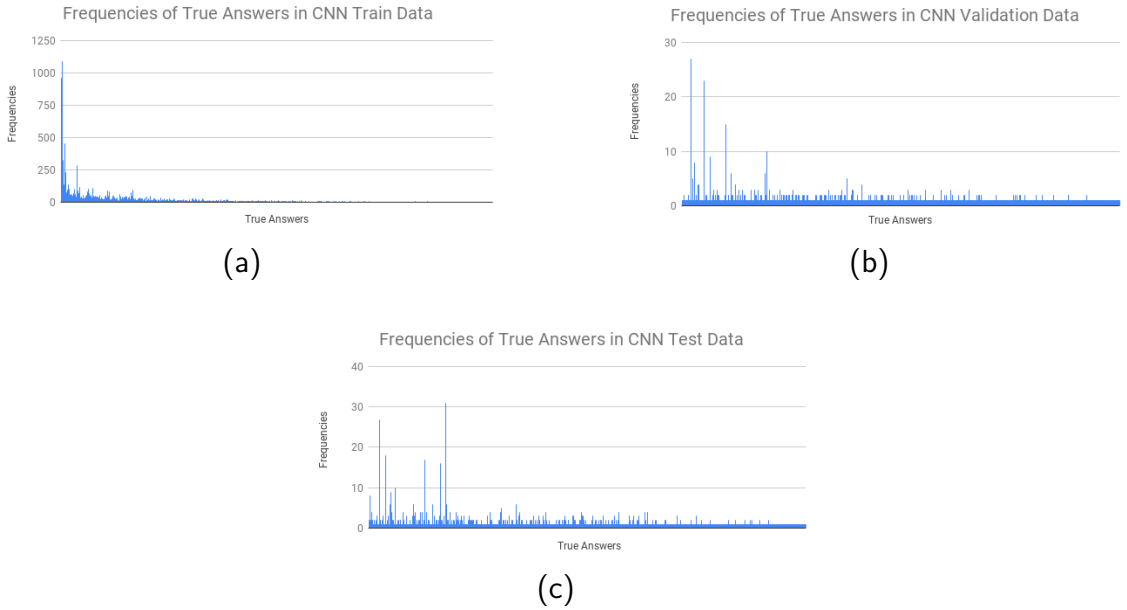

Figure B.3: Frequencies of true answers of CNN (a) train data, (b) validation data, and (c) test data.

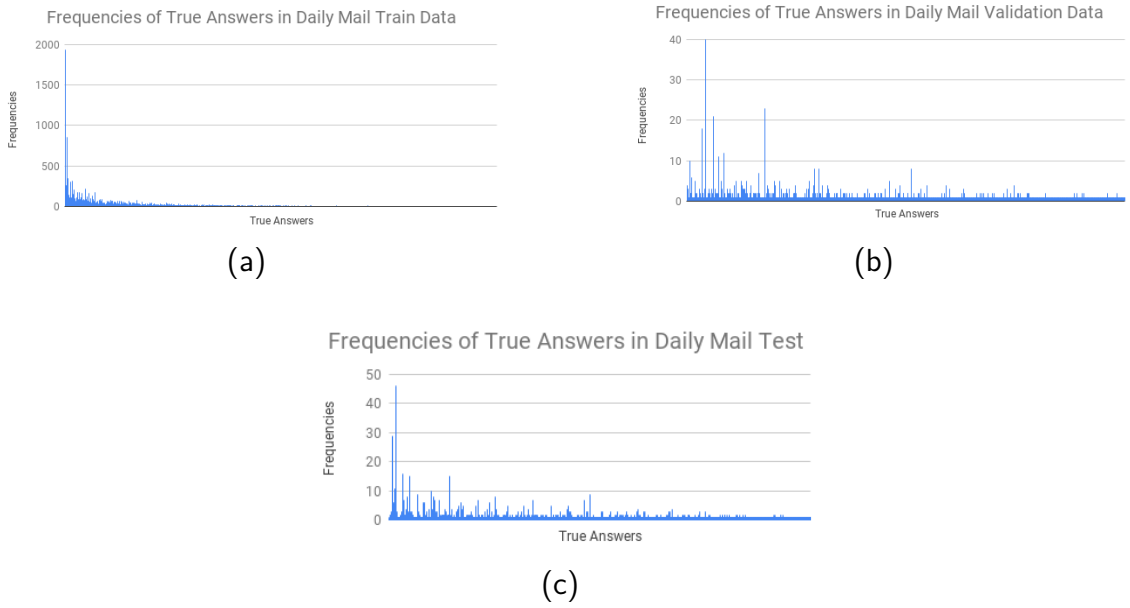

Figure B.4: Frequencies of true answers of Daily Mail (a) train data, (b) validation data, and (c) test data.

APPENDIX C

EXPERIMENTING WITH HYPER-PARAMETERS

Tables C.1 and C.2 illustrate results of experimenting with GRU and embeddings sizes respectively. We set embeddings to 200 while experimenting with GRU sizes and set GRU size to 384 while experimenting with embeddings. The experiments are conducted on CBT-CN data as this data was the one used by Alpay et al. [2].

Table C.1: Results of experimenting with GRU sizes.

GRU Sizes	**Test Accuracy**
256	64.1%
384	67.42%
512	62.41%

Table C.2: Results of experimenting with embeddings sizes.

Embeddings Sizes	**Test Accuracy**
100	64.82%
200	67.42%
300	64.86%

Based on recent reviewed work addressing cloze-style QA tasks using deep learning approaches presented in section 3.2, the most commonly used optimization algorithm is Adam [34]. Adam has several parameters to be set appropriately. The parameters are learning-rate and exponential decay rates of moving averages, namely, β_1 and β_2.

The default values of the parameters are 0.001 for learning-rate, 0.9 for β_1, and 0.999 for β_2 [34]. We measure effects of the parameters when the default values are used. Also, we experiment with two values less and two values greater than defaults. The justification of this decision is to measure effects of parameters when they are decreased and increased. When we experiment with a parameter, we set the other parameters to their default values. This is due to long training times (e.g., it takes a model several hours to converge), so it is not feasible to check all possible combinations. Table C.3 shows values of each of Adam's parameters.

Table C.3: Values of Adam's parameters.

Parameters	**Values**
learning-rate	[0.0001, 0.0005, 0.001, 0.005, 0.01]
β_1	[0.8, 0.85, 0.9, 0.95, 1.0]
β_2	[0.9, 0.99, 0.999, 0.9999, 1.0]

Table C.4 presents results of experimenting with Adam's parameters on validation and test sets of CBT-CN data. Default parameters result in 71.02% accuracy on validation set and 67.42% accuracy on test set. The table shows that default values of Adam's parameters give the best test accuracy. Learning-rate of 0.005 and β_1 of 1.0 result in unstable training. For instance, gradients become very small, and they are represented by Not a Number (NaN) values. As a result, parameters of recurrent

units get updated to NaNs.

Table C.4: Test results of experimenting with Adam's parameters. Results are expressed in accuracy scores.

Parameters	**Values**	**Results**	
		Validation Set	**Test Set**
Learning-rate	0.0001	67.59%	63.662%
	0.0005	71.67%	67.389%
	0.005	unstable	unstable
	0.01	60.43%	57.332%
β_1	0.8	68.95%	65.585%
	0.85	70.72%	67.388%
	0.95	72.18%	67.708%
	1.0	unstable	unstable
β_2	0.9	71.07%	67.268%
	0.99	71.77%	65.905%
	0.9999	70.87%	67.14%
	1.0	14.36%	14.183%
-	Default	71.02%	67.42%

REFERENCES

[1] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1480–1489. [Online]. Available: https://www.aclweb.org/anthology/N16-1174

[2] T. Alpay, S. Heinrich, M. Nelskamp, and S. Wermter, "Question answering with hierarchical attention networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[3] F. Hill, A. Bordes, S. Chopra, and J. Weston, "The goldilocks principle: Reading children's books with explicit memory representations," *arXiv preprint arXiv:1511.02301*, 2015.

[4] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proceedings of the 28th International Conference on Neural Information Processing Systems -*

Volume 1, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, pp. 1693–1701. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969239.2969428

[5] Y. Shen, P.-S. Huang, J. Gao, and W. Chen, "Reasonet: Learning to stop reading in machine comprehension," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1047–1055.

[6] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[8] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[10] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27.

[11] D. Ravichandran and E. Hovy, "Learning surface text patterns for a question answering system," in *Proceedings of the 40th Annual meeting of the association for Computational Linguistics*, 2002, pp. 41–47.

[12] E. Brill, S. Dumais, and M. Banko, "An analysis of the askmsr question-answering system," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 2002, pp. 257–264.

[13] V. Lopez, M. Pasin, and E. Motta, "Aqualog: An ontology-portable question answering system for the semantic web," in *European Semantic Web Conference*. Springer, 2005, pp. 546–562.

[14] F. Rinaldi, J. Dowdall, K. Kaljurand, M. Hess, and D. Mollá, "Exploiting paraphrases in a question answering system," in *Proceedings of the second international workshop on Paraphrasing*, 2003, pp. 25–32.

[15] S. Quarteroni and S. Manandhar, "Designing an interactive open-domain question answering system," *Natural Language Engineering*, vol. 15, no. 1, p. 73, 2009.

[16] A. Saxena, A. Tripathi, and P. Talukdar, "Improving multi-hop question answering over knowledge graphs using knowledge base embeddings," in *Proceedings of*

the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 4498–4507.

[17] Y. Deng, Y. Xie, Y. Li, M. Yang, N. Du, W. Fan, K. Lei, and Y. Shen, "Multi-task learning with multi-view attention for answer selection and knowledge base question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6318–6325.

[18] H. Jin, Y. Luo, C. Gao, X. Tang, and P. Yuan, "Comqa: Question answering over knowledge base via semantic matching," *IEEE Access*, vol. 7, pp. 75 235–75 246, 2019.

[19] M.-C. Yang, N. Duan, M. Zhou, and H. C. Rim, "Joint relational embeddings for knowledge-based question answering," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 645–650.

[20] W. Zhong, D. Tang, N. Duan, M. Zhou, J. Wang, and J. Yin, "Improving question answering by commonsense-based pre-training," in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2019, pp. 16–28.

[21] P. Liu, C. Zhang, W. Zhang, Z. Zhan, and B. Zhuang, "Attention-based memory network for sentence-level question answering," in *Chinese National Conference on Social Media Processing*. Springer, 2017, pp. 104–115.

[22] M. Li, X. Hou, J. Li, and K. Gao, "Superimposed attention mechanism-based cnn network for reading comprehension and question answering," in *Interna-*

tional Conference of Pioneering Computer Scientists, Engineers and Educators. Springer, 2019, pp. 25–37.

[23]

[24] V. Yadav, V. Bharadwaj, A. Bhatt, and A. Rawal, "Question–answer system on episodic data using recurrent neural networks (rnn)," in *Data Management, Analytics and Innovation*. Springer, 2020, pp. 555–568.

[25] A. Azzam, N. Tazi, and A. Hossny, "A question routing technique using deep neural network for communities of question answering," in *International Conference on Database Systems for Advanced Applications*. Springer, 2017, pp. 35–49.

[26] M. Oita, K. Vani, and F. Oezdemir-Zaech, "Semantically corroborating neural attention for biomedical question answering," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 670–685.

[27] H. Wu, M. Liu, J. Wang, J. Xie, and S. Li, "Question-answering aspect classification with multi-attention representation," in *China Conference on Information Retrieval*. Springer, 2018, pp. 78–89.

[28] R. Shrestha, K. Kafle, and C. Kanan, "Answer them all! toward universal visual question answering models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 10 472–10 481.

[29] R. Cadene, C. Dancette, M. Cord, D. Parikh *et al.*, "Rubi: Reducing unimodal biases for visual question answering," in *Advances in neural information processing systems*, 2019, pp. 841–852.

[30] M. Shah, X. Chen, M. Rohrbach, and D. Parikh, "Cycle-consistency for robust visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 6649–6658.

[31] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 6281–6290.

[32] L. Li, Z. Gan, Y. Cheng, and J. Liu, "Relation-aware graph attention network for visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 313–10 322.

[33] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] C. Fu, Y. Li, and Y. Zhang, "Atnet: Answering cloze-style questions via intra-attention and inter-attention," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2019, pp. 242–252.

[36] C. Fu and Y. Zhang, "Ea reader: Enhance attentive reader for cloze-style question answering via multi-space context fusion," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6375–6382, 07 2019.

[37] B. Dhingra, H. Liu, Z. Yang, W. Cohen, and R. Salakhutdinov, "Gated-attention readers for text comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1832–1846. [Online]. Available: https://www.aclweb.org/anthology/P17-1168

[38] R. Kadlec, M. Schmid, O. Bajgar, and J. Kleindienst, "Text understanding with the attention sum reader network," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 908–918. [Online]. Available: https://www.aclweb.org/anthology/P16-1086

[39] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[40] Y. Du, B. Pei, X. Zhao, and J. Ji, "Deep scaled dot-product attention based domain adaptation model for biomedical question answering," *Methods*, vol. 173, pp. 69–74, 2020.

[41] G. Balikas, A. Krithara, I. Partalas, and G. Paliouras, "Bioasq: A challenge on large-scale biomedical semantic indexing and question answering," in *Revised*

Selected Papers from the First International Workshop on Multimodal Retrieval in the Medical Domain - Volume 9059. Berlin, Heidelberg: Springer-Verlag, 2015, pp. 26–39. [Online]. Available: https://doi.org/10.1007/978-3-319-24471-6_3

[42] W. Li, W. Li, and Y. Wu, "A unified model for document-based question answering based on human-like reading strategy," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[43] X. Qiu, P. Qian, and Z. Shi, "Overview of the nlpcc-iccpol 2016 shared task: Chinese word segmentation for micro-blog texts," in *Natural Language Understanding and Intelligent Applications*, C.-Y. Lin, N. Xue, D. Zhao, X. Huang, and Y. Feng, Eds. Cham: Springer International Publishing, 2016, pp. 901–906.

[44] L. Xiao, N. Wang, and G. Yang, "A reading comprehension style question answering model based on attention mechanism," in *2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE, 2018, pp. 1–4.

[45] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. [Online]. Available: https://www.aclweb.org/anthology/D16-1264

[46] Y. Peng and B. Liu, "Attention-based neural network for short-text question answering," in *Proceedings of the 2018 2nd International Conference on Deep*

Learning Technologies, 2018, pp. 21–26.

[47] M. Wang, N. A. Smith, and T. Mitamura, "What is the Jeopardy model? a quasi-synchronous grammar for QA," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 22–32. [Online]. Available: https://www.aclweb.org/anthology/D07-1003

[48] L. Yang, Q. Ai, J. Guo, and W. B. Croft, "anmm: Ranking short answer texts with attention-based neural matching model," in *Proceedings of the 25th ACM international on conference on information and knowledge management*, 2016, pp. 287–296.

[49] J. Bao, N. Duan, M. Zhou, and T. Zhao, "Knowledge-based question answering as machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 967–976.

[50] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on freebase from question-answer pairs," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1533–1544.

[51] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

[52] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.

VITAE

- Name: Fahad Alsahli
- Nationality: Saudi
- Date of Birth: 29 Aug 1994
- Email: *fahad@fahadsahli.com*
- Website: *https://fahadsahli.com*
- Permenant Address: Riyadh 13314 - 8247, Kingdom of Saudi Arabia