Would you like fries with that? Modular Multi-hop Reasoning

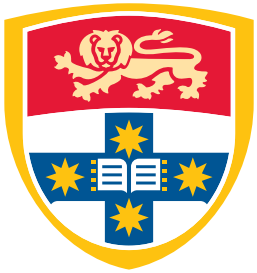
HAMISH J. IVISON
SID: 460299200

Supervisor: Dr. Caren Han
Associate Supervisor: Dr. Josiah Poon

This thesis is submitted in partial fulfillment of
the requirements for the degree of
Bachelor of Information Technology (Honours)

School of Computer Science
The University of Sydney
Australia

18 November 2020



THE UNIVERSITY OF
SYDNEY

Student Plagiarism: Compliance Statement

I certify that:

I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure;

I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to the University commencing proceedings against me for potential student misconduct under Chapter 8 of the University of Sydney By-Law 1999 (as amended);

This Work is substantially my own, and to the extent that any part of this Work is not my own I have indicated that it is not my own by Acknowledging the Source of that part or those parts of the Work.

Name: Hamish J. Ivison

Signature: **Date**: November 18, 2020

Abstract

In this work, we investigate an interpretable, modular approach to multi-hop question answering by adapting a popular visual question answering architecture, the MAC cell, to the task of multi-hop reading comprehension. In multi-hop reading comprehension, a model must answer questions by collating facts from multiple text sources. Our augmented MAC cell design outperforms existing modular approaches to multi-hop QA with less supervision and provides interpretable insights into its reasoning process. We then investigate integrating our cell with the highly popular BERT model and design a novel model which iteratively reads and retrieves documents in an interpretable fashion, allowing scalable and interpretable multi-hop question answering. Alongside this, we investigate the behaviour of generic BERT-based models on multi-hop QA and show that several existing approaches to multi-hop QA fail to significantly beat a naive BERT baseline. Our work shows the promise of MAC networks for multi-hop reasoning and outlines future paths for both MAC networks and multi-hop reasoning as a whole.

Acknowledgements

It has been a long and extraordinary year in the truest sense of the word, and I have many people to thank for the fact that I have been able to produce this thesis and finish my undergraduate degree. First and foremost, I must thank my parents, who have provided me with endless care and support throughout my life, and especially throughout this year. It is no exaggeration that this thesis would have not been completed without their continued support and care throughout this turbulent year. Secondly, I must thank my supervisor and co-supervisor, Dr Caren Han and Dr Josiah Poon. Their guidance and feedback on my research has been excellent and will shape the way I approach problems in the years to come. I especially thank Dr Han for her continued patience with many meetings involving last-minute bug fixes or research side-tangents and her continued belief in my work. Thirdly, I would like to thank the other members of the USYD NLP research group, to whom I have given several talks, and after each one have provided useful and fresh feedback. Finally, I would like to thank the Commonwealth Bank of Australia, Appen, and USYD for their financial support of my studies throughout my degree through several scholarships. In many ways, this aid has made my time at university far more enjoyable and successful than it otherwise could have been.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Question Answering and Reading Comprehension

One of the oldest tasks in the field of natural language processing (NLP) is that of question answering (QA), dating back to Alan Turing and the Turing test (Turing, 1950). In QA, systems are tasked with generating (usually natural language) responses to natural language questions. While the scope of potential questions is large, traditionally QA research focuses on answering 'factoid questions' - questions that can be answered with a single fact from a (usually short) snippet of text. Such systems have a wide variety of use-cases, from aiding internet search (Hazen, 2019; Nayak, 2019), to winning quiz shows (Gustin, 2017; Markoff, 2011; Ferrucci et al., 2010), to providing answers to medical questions (Möller et al., 2020). Despite the success of these recent QA systems, there is still much space for improvement in various directions, with current models struggling on particularly complex or large-scale QA tasks (Dua et al., 2019; Yang et al., 2018). As such, QA is a highly interesting area of research with obvious real-world applications and many potential directions for future research.

In this work, we focus on a subset of QA called *multi-hop machine reading comprehension*. In machine reading comprehension (MRC), a model is presented with passages from a (or multiple) text(s) and is then asked questions that test its understanding of the text, similar to the reading comprehension exams found in primary and secondary education. These questions can range from asking for simple facts found in the text (e.g. 'Who is the CEO of Apple?') to more complex questions involving different types of reasoning (e.g. 'Who died first: Ferdinand II or Charles V?' or 'How many empires attacked Guadalajara?'). Questions that require drawing information from multiple input texts are called *multi-hop* questions. These questions have received much study recently, with the rise of multi-hop specific datasets (Yang et al., 2018; Welbl et al., 2018) and models tackling these datasets (Ding et al., 2019; Fang et al., 2020; De Cao et al., 2019). Existing work on these questions largely focuses on either (a) constructing complex graphs for information passing between various condensed representations of

portions of the input text, or (b) improving the ability of the system to narrow down the input text to just the relevant facts needed for answering. In contrast, we investigate adapting an attention-based model to this task, which does not rely on graphs while still containing a strong inductive bias for multi-hop QA. In addition, we investigate the importance of narrowing down the input text to existing models and present a new model for doing so in a scalable and interpretable manner.

1.2 Contributions

We tackle the task of multi-hop machine reading comprehension by adapting a novel network design, the MAC cell (Hudson and Manning, 2018) to the task of multi-hop reading comprehension. Our adapted design provides improvements over existing modular neural network approaches to a popular multi-hop QA dataset. We provide a detailed analysis of our model, showing it contains a strong inductive bias for multi-hop reasoning, and investigate how well the interpretable qualities of MAC cells are maintained when operating over text instead of over images (for which they were originally designed). We then augment our model with the popular BERT model (Devlin et al., 2019) and explore how this integration changes the impact of the MAC cells. While a naive application of the MAC cell provides little benefit over a baseline BERT model, we show that certain elements of MAC cells provide clear utility for document selection in multi-hop QA. Finally, we show that competitive performance on the popular multi-hop dataset HotpotQA can be achieved without performing cross-document reasoning, highlighting the weaknesses of the HotpotQA dataset and providing insight into the answering strategies used by current state-of-the-art multi-hop QA models.

In summary, we make 4 key contributions:

(1) We adapt the MAC network to machine reading comprehension, bringing a popular image-based model to a text-based task. We provide a detailed analysis of the behaviour of this novel model design.
(2) We show that the adapted MAC network provides stronger or competitive performance compared to existing modular approaches, highlighting the effectiveness of the augmented MAC cell for multi-hop reasoning.
(3) We show that cross-document reasoning is largely not required for state-of-the-art performance on the popular multi-hop QA dataset HotpotQA and that most performance gains come from the relatively under-investigated document selection step.

(4) We design a multi-hop model that achieves competitive performance on HotpotQA by focusing on the document selection step. This model achieves performance on par with existing models with dedicated reader components without using such a component by concurrently selecting documents while answering a given question.

Note that we have proposed *two* different, but related, models above:

(1) A model that serves as a basic adaption of the MAC cell to the task of multi-hop reasoning. As this model utilises GloVe-based embeddings to represent text, we henceforth refer to this as our **GloVe-based model**.
(2) A model that integrates the BERT model for representing input text. Integrating BERT is essential for state-of-the-art performance, but requires further modifications to our design. We henceforth refer to this as our **BERT-based model**.

These two models provide two different contributions: (1) highlights the promise of MAC cells when applied text-based QA, and (2) highlights how MAC-cell based methods can be best integrated with BERT-style models to make use of both the strengths of BERT and MAC cells.

1.3 Structure

In this introduction, we have outlined the task of multi-hop question answering and summarised the core contributions of this work. We now lay out the structure of the rest of the work:

In **chapter 2**, we provide a brief history of QA and reading comprehension before then covering recent popular neural approaches to reading comprehension-style QA. We then cover the state-of-the-art in multi-hop QA and conclude by covering other work adapting MAC cells to text-based reasoning tasks. We provide summary tables for all models discussed.

In **chapter 3**, we provide a detailed analysis of the datasets used for evaluation and describe the baseline models we compare our models against. We also set out the core methods used to evaluate our models.

In **chapter 4**, we describe the design of model (1) above in detail.

In **chapter 5** we present a detailed qualitative and quantitative analysis, including hyperparameter tuning, ablations, performance evaluation, and interpretability analysis. We then examine a naive integration of BERT with model (1) and show that current approaches to HotpotQA rely heavily on a little-examined document selection step.

In **chapter 6** we describe the design of model (2) above, motivated by our findings in the previous chapter.

In **chapter 7**, we present an analysis of model (2)'s performance similar to analysis performed in chapter 5.

Finally, in **chapter 8** we discuss future directions for the area of multi-hop reading comprehension and our work, before then concluding by summarising the results and contributions of this work.

CHAPTER 2

Literature Review

2.1 Introduction

In this section, we provide a brief history of question answering in NLP and then perform a thorough review of recent progress made in the area of multi-hop reading comprehension. We first examine highly-popular most popular reading comprehension models: BiDAF (Seo et al., 2017), DrQA (Chen et al., 2017), and BERT (Devlin et al., 2019). These models have been highly influential and aspects of their designs are still largely used in both QA and the field of NLP as a whole. We then examine current trends in the state-of-the-art in multi-hop reasoning, focusing on progress made on HotpotQA (Yang et al., 2018), a highly popular multi-hop QA dataset. Finally, we review other attempts to extend MAC-related networks to textual reasoning.

2.2 A Brief History of QA

Early systems for QA, such as BASEBALL (Green et al., 1961) and LUNAR (Woods, 1977) focused on a *parsing-based* approach to QA, mapping natural language sentences to structured database queries, and were only capable of answering questions about single domains: BASEBALL answered questions about baseball players, while LUNAR answered questions about lunar rocks. While most initial work on QA followed this parsing-based approach, there was a small amount of work done on building systems that could *reason* over text, an early example being the QUALM system (Lehnert, 1977). Such work took an *information retrieval* approach to QA, where the system itself learnt to retrieve facts directly from an underlying text, rather than from a structured knowledge base. This work eventually led to the development of the first machine reading comprehension task (Hirschman et al., 1999), which drew upon reading comprehension questions from primary school exams for its data. Initial approaches to this task before 2015 largely relied on fragile and labour-intensive systems utilising hand-crafted rules

(Riloff and Thelen, 2000; Ng et al., 2000) and/or simple word-based heuristics (Hirschman et al., 1999; Charniak et al., 2000). Even with these hand-crafted approaches, early work struggled even on basic reading comprehension tests. As such, reading comprehension was under-examined for some time due to its seeming immense difficulty as a task.

However, around 2013-2015 the amount of research into the reading comprehension task exploded due to two factors: firstly, the construction of large reading comprehension datasets, such as MCTest (Richardson et al., 2013), SQuAD (Rajpurkar et al., 2016), and CNN/Daily Mail (Hermann et al., 2015), and secondly, the rise of neural network-based approaches to reading comprehension (Hermann et al., 2015; Weston et al., 2015; Yin et al., 2016; Kadlec et al., 2016; Cui et al., 2016). These datasets formulated the reading comprehension task as either a location problem, where the model had to locate the correct answer to a question within some input text, or as a multiple-choice problem, where the model had to select the correct answer from a list given a question and input text. Neural approaches proved incredibly effective for these tasks, which themselves served as excellent tests for the 'general' reasoning ability of these models. Successful neural approaches to these newer datasets largely relied on attention-based mechanisms, with notable examples being the bi-attention flow model (Seo et al., 2017) and BERT (Devlin et al., 2019), with recent approaches achieving above-human performance (Devlin et al., 2019; Lan et al., 2020). We cover some of these approaches below. This large success has inspired a more recent wave of newer reading comprehension tasks which further increase the difficulty of the task either by requiring question answering over longer passages of text (Kwiatkowski et al., 2019), or by requiring varied and difficult forms of reasoning (Khashabi et al., 2018; Dua et al., 2019), or both (Yang et al., 2018; Welbl et al., 2018).

One particular method for increasing the difficulty of reading comprehension has been the incorporation of multi-hop questions - questions which require examining facts from multiple documents in order to answer. Such questions simultaneously require reasoning over longer contexts than prior reading comprehension tasks and require more complex reasoning due to the requirement of multiple facts to locate the answer. Recent models that tackle these more difficult reading comprehension tasks often use several shared core features: a retriever model to winnow down the passages of text to contain only sentences relevant to the question (Asai et al., 2020), a contextual word embedding model for constructing information-rich word representations (e.g. BERT (Devlin et al., 2019)), and an output layer or module for making the final answer prediction. We cover several recent approaches to multi-hop QA below.

2.3 Attention-based Machine Reading Comprehension

As mentioned above, early successful neural approaches to reading comprehension largely focused on the use of attention, with a recurrent neural network used to process the text and an attention mechanism used to determine the most relevant parts of the input text to the question. As attention mechanisms to this day remain a core part of most NLP models, we now provide a brief description of what 'attention' means in the context of NLP and reading comprehension.

Intuitively, attention mechanisms calculate what words or items in a sequence are most 'important' to a given word or item, and use these to construct a context vector summarising the sequence weighting the more 'important' items in the sequence more heavily. In its most simple form, we have some sequence u and item h we wish to calculate attention against (often u is a sequence of word embeddings, and h a single embedding). We first calculate a score between each item of u and h:

$$u = [u_1, u_2, ..., u_t] \tag{2.1}$$

$$A_i = \text{score}(u_i, h), i = 1...t \tag{2.2}$$

This score function can take many forms, with the currently most popular function being the dot product (Vaswani et al., 2017). After this, we convert the scores to probabilities using the softmax function and use the probabilities to construct a summary vector of u:

$$A' = \text{softmax}(A) \tag{2.3}$$

$$u_s = \sum_{i=1}^{i=t} A'_i u_i \tag{2.4}$$

This summary vector is simply a weighted sum of items in u, and so holds more information from the items in u deemed most important to h. This summary vector can then be used in a variety of ways depending on the task of a given model. In addition, multiple attention mechanisms are often combined in various ways, including bi-attention, where attention summary vectors are produced for two sequences at once (Seo et al., 2017), and multi-head attention, where multiple attention mechanisms are applied in parallel on the same underlying data (Vaswani et al., 2017).

2.4 Neural Approaches to QA

We now go over three highly-influential neural-based approaches to reading comprehension. Each approach introduces ideas and elements that are core parts of many state-of-the-art reading comprehension models. All approaches take in a question and input document(s) and find the answer to the question within the text of the input document(s).

2.4.1 Bi-directional Attention Flow Model

The bi-directional attention flow model (BiDAF) (Seo et al., 2017) is a highly influential reading comprehension model, introducing the highly useful bi-attention layer. This model first takes in a question and context (i.e. the input text) in text form, and turns them into vectors using both GloVe word embeddings[1] (Pennington et al., 2014) and learnt character embeddings. These embeddings are combined using a bidirectional LSTM layer[2] (Hochreiter and Schmidhuber, 1997), constructing contextually-aware vector representations of each token in the context and question, allowing the model to implicitly model the syntax and semantics present in the underlying text. A novel attention mechanism, the attention flow layer (or bi-attention layer) is then applied, which calculates interactions between the question and context[3]. These interactions are passed through another set of LSTM layers (the 'modelling layer'), and then a final set of LSTM layers are used to predict the location of the answer in the context. A diagram of this architecture is given in figure 2.1.

The proposed bi-attention layer is now a widely-used component in QA models (Qiu et al., 2019; Fang et al., 2020), and allowed the BiDAF model to achieve state-of-the-art results on a set of QA datasets, including the popular SQuAD dataset (Rajpurkar et al., 2016), at its time of publishing. However, its reliance on LSTM layers limits its ability to model long-distance dependencies, and the model struggles with multi-step reasoning due as it only calculates interactions between the question and context once in its bi-attention layer (Jiang and Bansal, 2019a). Furthermore, the bi-attention mechanism has limited interpretability, especially when applied to large inputs, as it requires visualising a value for every word-pair between the question and context.

[1] When we refer to 'X embeddings' here and throughout this work, we mean dense vector representations of X, often learnt through training or specific embedding techniques such as GloVe.

[2] A popular recurrent neural network, see chapter 4 for more details.

[3] We provide more details on the bi-attention layer in chapter 4

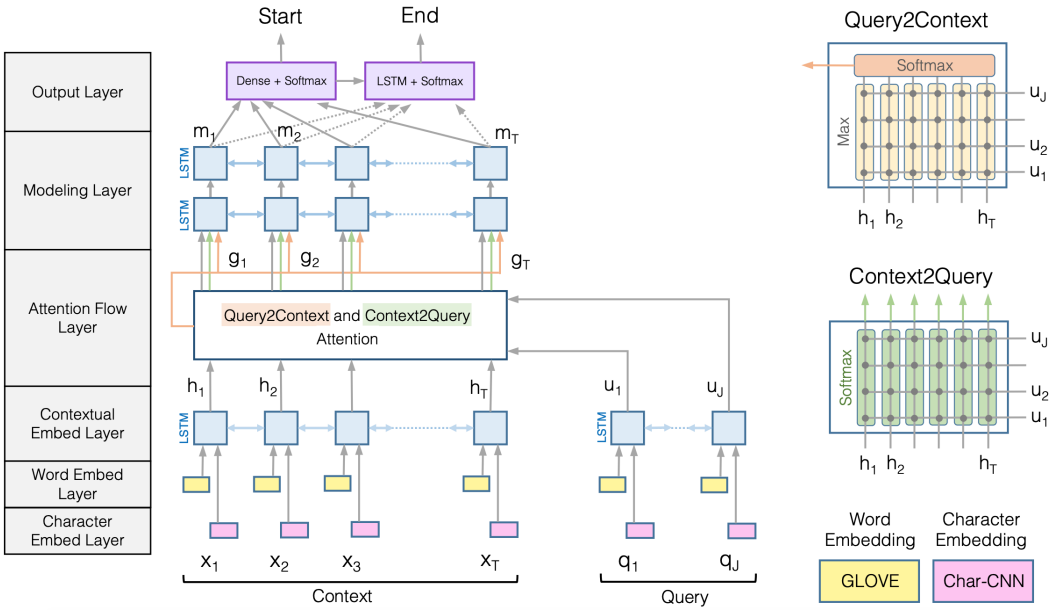


FIGURE 2.1. Diagram of BiDAF model from Seo et al. (2017). Query2Context and Context2Query represent the attention mechanisms between the context and query.

2.4.2 DrQA

Another popular attention-based model is DrQA (Chen et al., 2017), which tackles the task of open-domain QA, where a model must find relevant document text from (up to) millions of documents first before finding the specific answer to a question. DrQA is a pipeline model, consisting of two components: a document retriever and a document reader. The document retriever first identifies documents relevant to a given question using traditional information retrieval techniques, including TF-IDF scores and n-gram features (see Jurafsky and Martin, 2009, chap. 23). The top 5 ranked documents are then independently fed into a document reader model, which predicts an answer in each document, and the most likely answer overall is chosen as the final answer. The document reader model is a simple model using LSTM layers and multiple input features for each word, but the overall pipeline architecture itself is agnostic to the specific reader model design.

DrQA achieved state-of-the-art results at the time of publishing, beating the BiDAF model on the traditional SQuAD setting, whilst also achieving good results on open-domain datasets where it is required to locate answer documents from a set of thousands in addition to predicting the precise answer to a question. The 'retrieve and then read' paradigm proposed in the DrQA model is now the current standard across open-domain and long-document reading comprehension, although the models used for reading

and retrieving have changed. This is due to the basic nature of the two models proposed: utilising more complex neural methods for document retrieval can aid in retrieving documents with little lexical overlap with a given question. In addition, as each document is processed independently, DrQA is unsuitable for datasets such as HotpotQA which require sharing information across documents to find the answer.

2.4.3 Pretrained Language Models

Recently, pretrained language models have overtaken the field of NLP, including QA, due to their high performance and versatility across many different NLP tasks. These models are largely based on the model proposed in Devlin et al. (2019), commonly referred to as 'BERT'. These models utilise large stacks of attention mechanisms and lengthy pre-training tasks on vast amounts of data to learn how to produce rich contextual vector representations of words in an unsupervised manner. After pre-training, these models are then 'fine-tuned' for a specific dataset or task by training for a few epochs on the given dataset/task. This allows such models utilise general language understanding abilities learnt through pre-training for these specific tasks, and such an approach has yielded state-of-the-art performance for various reading comprehension datasets (Devlin et al., 2019; Yang et al., 2019). We give a more thorough description of the architecture of BERT and related models in chapter 6.

The power of these models lies in their size and pretraining: they are extremely large models (the 'base' version of BERT having over 100 million parameters), with excellent pattern memorisation abilities, and through pre-training on large amounts of language, can gain an impressively deep syntactic and semantic understanding of language. However, these models are extremely computationally expensive due to heavy reliance on self-attention mechanisms, which require the calculation of scores between every pair of words in an input text, resulting in $O(n^2)$ runtime and significant GPU memory requirements. Due to this, these models can only process texts of relatively short length, with the task of improving their scalability being an ongoing research question. As such, these models are still a long way off being able to process anything more than a handful of documents at once due to computational and memory constraints. In addition, recent work has shown that there is room for improvement on these models in multi-hop question answering through the addition of specialised layers applied to their outputs (Wang et al., 2019c), although the potential gains are small. Finally, these models are almost entirely black boxes, with attempts to understand their internals requiring sophisticated probing experiments (Tenney et al., 2019). Thus, while large pretrained language models are certainly important components of

the current state-of-the-art QA methods and have achieved impressive results, there is still space for improvement on these models in several areas.

These three popular models serve both as components of and inspiration for the current state-of-the-art in multi-hop QA, and we leverage aspects of their designs for our own models. We summarise these models in table 2.1 below.

Paper	**Model**	**Dataset(s)**	**Task**
Seo et al. (2017)	BiDAF	SQuAD, CNN/Daily Mail dataset	Extractive QA
Chen et al. (2017)	DrQA	SQuAD, WebQuestions (Berant et al., 2013), CuratedTREC (Baudiš and Šedivý, 2015), WikiMovies (Miller et al., 2016)	Open-domain extractive QA
Devlin et al. (2019)	BERT	SQuAD, GLUE (Wang et al., 2019b)	Extractive QA, Natural language understanding

TABLE 2.1. Summary of models discussed in section 2.3.

2.5 Multi-Hop Question Answering

While the models discussed above are extremely popular and influential, they were originally largely developed for basic reading comprehension setups, where the model is given a small segment of text and tasked with finding an answer to a question in the text, with the prior knowledge that there is always a valid answer present in the text (the exception being DrQA, as discussed above). This constrained version of the task is largely now 'solved', with superhuman performance achieved on popular datasets using this setup (e.g. SQuAD). As such, recent work has examined developing more challenging and useful reading comprehension tasks. One such work is the HotpotQA dataset (Yang et al., 2018), which shares the same basic setup noted above, but increases the difficulty in two ways: first, by **asking questions that require multiple steps of reasoning** (called 'multi-hop reasoning'), and second by **providing longer segments of text**, requiring a model to not only deal with longer and noisier inputs but also to retrieve multiple facts and perform multi-step reasoning in order to find the correct answer. In the closed-domain *distractor setting* of HotpotQA, models are given 10 segments of text (usually paragraphs from Wikipedia) for each question, only 2 of which are required to find the answer. In the open-domain *full-wiki setting*, a model must instead find the relevant text segments from an entire dump of Wikipedia[4]. In both settings, models are additionally tasked with marking what sentences support their prediction

[4]A specific cleaned dump of Wikipedia is used to ensure the nature of the dataset does not change over time with Wikipedia.

(called 'supporting facts'), and must also be able to answer polar (or 'yes/no') questions. We provide more details on this dataset in chapter 3. In this section, we examine the current state-of-the-art across both HotpotQA setups, as this dataset is the primary focus of our work.

2.5.1 Closed-domain Multi-hop QA

One relatively early approach to the hotpot distractor setting is the dynamically fused graph network (DFGN) (Qiu et al., 2019). This network utilises a BERT model for document selection and then constructs a graph of entities to reason over the input documents. First, the 10 input documents are scored by a trained BERT model, and only documents over a certain score are selected (this score is tuned to maximise recall), as BERT is unable to process all 10 at once. The selected documents are simply concatenated into one long string and entities are extracted using the Stanford coreNLP toolkit (Manning et al., 2014). These entities serve as nodes in a graph, and edges are constructed between entities in the same sentence, entities with the same text, and entities in the same paragraph (via a constructed paragraph node). The concatenated text is then passed through a BERT model and bi-attention layer (Seo et al., 2017) along with the given question to construct contextual embeddings for the input documents. Importantly, this BERT model is *not* trained with the rest of the network, which greatly harms the performance of this network. These inputs are then passed through a series of 'fusion blocks', which iteratively construct embeddings for each node in the entity graph from word embeddings, mask out entities unrelated to the current reasoning step using a mask generated from the question embeddings, and then apply a graph attention layer (Veličković et al., 2018) to the entity graph, before finally using a bi-attention layer to update the word embeddings with information from the output of the graph attention layer (wherein information is passed between adjacent entity nodes). Each fusion block thus performs one reasoning step, each block focusing on a sub-part

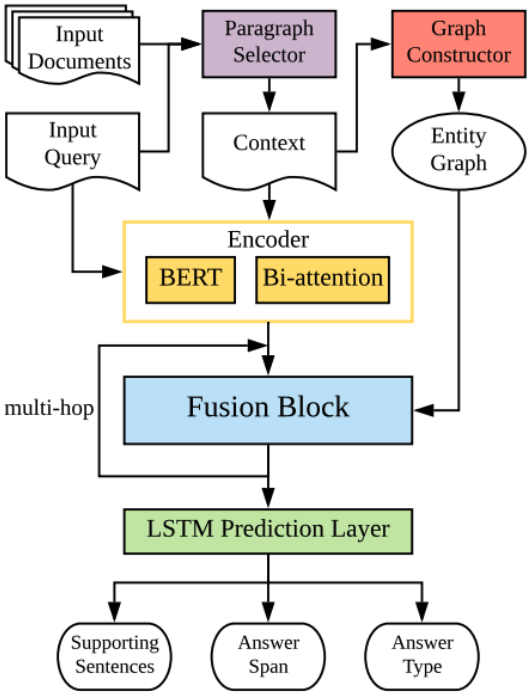


FIGURE 2.2. Diagram of the full DFGN pipeline from Qiu et al. (2019)

of the question. In order to perform answer and supporting fact prediction, a stack of LSTM layers is used, with each layer making one prediction and feeding its predictions into the next layer. We provide a diagram of the overall model in figure 2.2.

This model performs well above the non-BERT-based baseline, and ablations performed by the authors suggest the graph network is vital to its good performance. However, recent work (Shao et al., 2020) has found that fine-tuning the BERT model (not performed by the original DFGN network) and removing the fusion blocks can provide performance well above the complex DFGN model, throwing into doubt the need of the complex graph setup. This highlights the power of BERT for question answering and shows that attention-based layers such as the transformer (which makes up BERT) are more than suitable enough for multi-hop QA. Despite this, the use of iterative blocks for multi-step reasoning does provide benefits when not fine-tuning BERT, and also provides a more interpretable model, as one can examine the attention masks in each block to see what entities are being examined by the model. Thus, while overly complex, the sequential block-based design of the DFGN is still worth examining, especially as its iterative cell design shares some similarities with that of the MAC cell.

Tu et al. (2020) show the utility of BERT for QA by using a basic BERT model for answer prediction, along with more complex document selection and supporting fact models. This model, called the 'Select, Answer and Explain' (SAE) model, improves upon the basic document selection method of the DFGN by considering inter-document interactions, rather than scoring each document independently. It first constructs document vectors by passing each document (along with the question) through a BERT model and extracting the 'CLS' (or 'classification') token embedding from the output, which is a dense vector representation of the input document and question. The CLS representations from each document are then passed through a multi-head attention mechanism (the attention mechanism used in the transformer network, and discussed in more detail in chapter 6) to model interactions between each document, and then finally passed through a bilinear layer to provide a score for each document pair (D_i, D_j). The ground truth score is 1 if D_i is ranked higher than D_j and 0 otherwise, where the answer document is ranked first, other supporting document ranked second, and all other documents ranked equal third. During inference, the documents with the top two scores are chosen. We provide a diagram of the document selection model in figure 2.3. This method provides higher recall and accuracy on selecting the relevant documents than a simple BERT ranking approach (as was used in the DFGN) and makes use of past research into ranking for information retrieval (Liu, 2011). Given the top two scoring documents, the answer is then predicted by concatenating the two documents into a single string, and passing them

through a BERT model (as described above) along with the question. A graph composed of sentence embeddings linked through entity matching is constructed and passed through a multi-relational graph convolutional network (De Cao et al., 2019) to predict supporting facts and if the answer is yes or no.

This model achieved state-of-art results at its time of published, and outperformed the more complex DFGN model, highlighting the power of a well-finetuned BERT model. Its document selection method is far more accurate than the DFGN, and can easily be used in other approaches to the distractor setting. However, the gains reported by the use of the graph network for supporting facts are small, and we find later in this work we can outperform the SAE with far simpler approaches to supporting facts, suggesting again that these graph-based approaches are potentially unnecessary. In addition, while the heavy reliance on BERT is core to performance, the interpretability of the model is harmed due to its black-box nature, with the authors only showing some basic attention maps for the sentence embedding creation process (for which the simpler self-attention mechanism is easy to visualise). Finally, the pipeline nature of the SAE model means if the document selection is incorrect, the rest of the model is unable to 'course correct' and instead will always provide the wrong answer. As such, the SAE model provides an effective model for document selection and highlights the importance of BERT for QA, but is also potentially overly complex due to its use of graph networks with little performance gain.

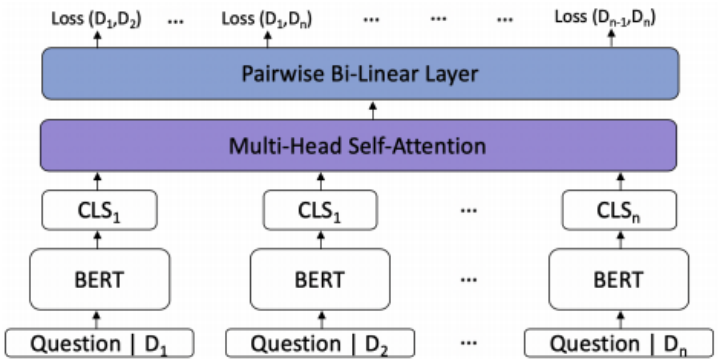


FIGURE 2.3. A diagram of the SAE document selection process from Tu et al. (2020). D_i refers to the text of the i^{th} document.

More recently, the hierarchical graph network (HGN) makes better use of graph networks for HotpotQA, being the current highest published model on the HotpotQA distractor leaderboard[5]. This model more tightly integrates a graph network into its reasoning process, adding a strong multi-hop inductive bias into the model. First, the HGN uses a BERT-based model to rank input paragraphs, like the DFGN. The top-two ranking documents are selected, and the top-two ranked documents hyperlinked to these first

[5]At the time of writing, November 18, 2020.

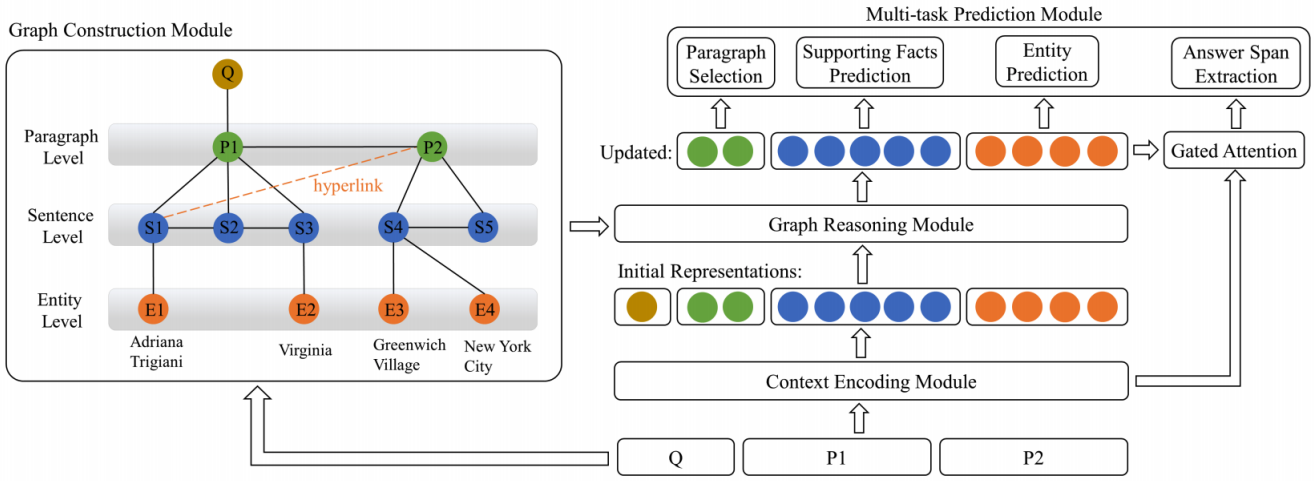


FIGURE 2.4. Diagram of HGN model from Fang et al. (2020)

two are then selected. This ranking makes use of the fact that questions within HotpotQA were created by examining Wikipedia hyperlinks, with supporting documents always connected by hyperlinks.

These four selected documents are then concatenated together with the question and passed through a BERT-based model along with a bi-attention layer to construct contextual question-aware word embeddings. These are then used to construct a hierarchical graph, where each node represents either the question, a paragraph, a sentence, or an entity in a sentence. The nodes are constructed by passing a bidirectional LSTM over the word embeddings and concatenating its hidden states at the start and end of each respective span of text. Entity spans are detected using the SpaCy library (Honnibal and Montani, 2017). The question node is constructed by max-pooling over the question word embeddings. Each node is then connected any span of text it is contained within (i.e. an edge is drawn from a paragraph node to all sentences inside it), as well as edges being added to connect hyperlinked documents and matching entity nodes. The final graph is passed through a graph attention network (Veličković et al., 2018), and the resulting enriched node representations are merged back with the contextual word embeddings using a gated attention mechanism. These graph-enriched word embeddings are then used to predict the answer, with the node representations used to predict supporting facts, as well as relevant entity and paragraph nodes (which are used in training to aid the model's reasoning process). We provide a diagram of the overall model in figure 2.4.

The HGN outperforms the SAE model, and at time of publishing was state-of-the-art on HotpotQA. Furthermore, ablation experiments show its graph usage improves on a non-graph model by 3 points, providing strong evidence the graph is useful in this case, although the majority of the edges provide

relatively small performance improvements, suggesting the graph could be pruned while retaining its original performance. However, the authors do not test utilising transformer networks in place of the graph attention network, which Shao et al. (2020) showed was possible to do for the DFGN model without harming results. Furthermore, the HGN's hyperlink-based document selection process is potentially exploitative of a bias in the HotpotQA dataset, which was constructed using hyperlinks, limiting its application to other datasets and tasks. Finally, similarly to the SAE model, the fact that document selection is a separate step means the model is unable to recover from an erroneous document selection step. Thus while the HGN shows an effective method for integrating graph-based reasoning more tightly than the SAE model, its graph usage is still potentially overly complex, and its document selection step could be further improved.

As we have seen, closed-domain multi-hop QA models are quick to use graph-based techniques for determining supporting facts and performing entity-based reasoning, but such complex graph techniques are potentially unnecessary with proper fine-tuning. Truly core to these methods is heavy usage of a BERT-based model and their document selection approach, as well as tighter integration of supporting facts into the answer prediction process. We focus on this document selection step in our second model, whilst also exploring non-graph based models that still provide strong multi-hop reasoning abilities.

2.5.2 Open-domain Multi-hop QA

In open-domain QA, tested by the HotpotQA full-wiki setting, models must retrieve relevant documents from thousands (or more) or documents in order to find an answer. This is a more realistic setting than the distractor setting, where models get a shortlist of documents guaranteed to contain the correct answer. As a result, models for the full-wiki setting focus more on document selection and retrieval than document reading, in contrast to the closed-domain distractor setting.

Ding et al. (2019) present a framework which iteratively constructs a knowledge graph from input documents called CogQA. It is comprised of two systems: first, a BERT model, which extracts entities and information from a document's text, slowly building a connected graph of entities, which store with them the sentence they appear in. Given an entity mention in a document, the BERT model predicts the likelihood of that entity being useful as the next step in the path or being the answer. If it is the next step, then a graph edge is constructed from the current entity's sentence to the paragraph with title matching that entity, with the paragraph itself retrieved and passed through the BERT model to determine the next

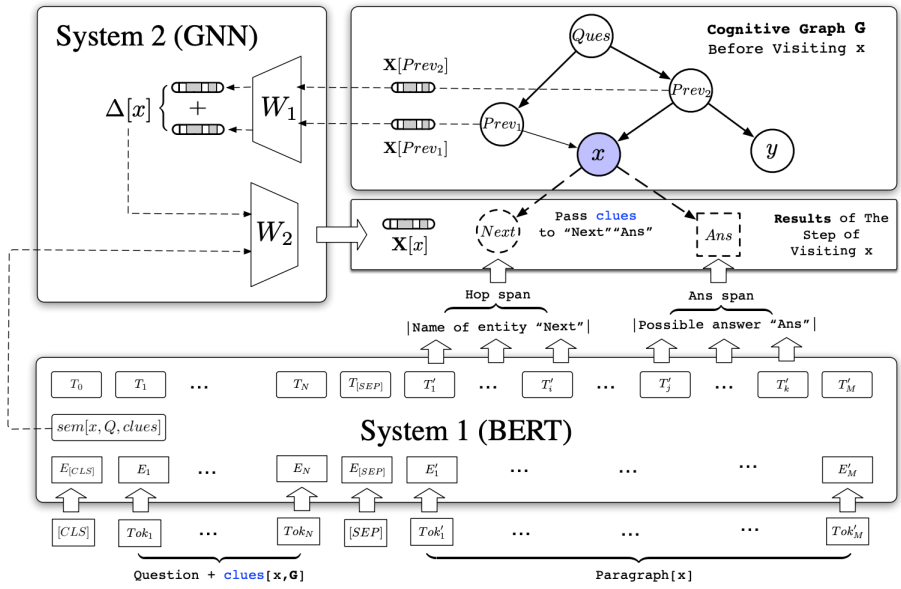


FIGURE 2.5. Diagram of proposed CogQA architecture from Ding et al. (2019), representing the step when visiting node x. $X[t]$ represents the hidden state of the graph at step t. Classification module not shown.

step. If it is an answer, an answer node is appended to the graph. The second system is a custom variant of a graph convolutional network (GCN) (Kipf and Welling, 2017), which the constructed graph is passed through to allow neighbouring nodes to share information and construct an encoded representation of the graph. The encoded representation of the final graph is then passed to a classification module to make a final prediction. An overview of this architecture is given in figure 2.5.

CogQA improved on the existing state-of-the-art when evaluated on HotpotQA's full-wiki setting, with especially drastic improvements in the retrieval of supplementary facts (which are output by examining the entities the model uses to build the graph). Furthermore, the authors also note that by examining the constructed graph, the reasoning steps made by the model can be determined, adding some degree of interpretability. The authors also show the proportion of answers with correct explanations is higher for CogQA than previous models, supporting their claim that CogQA is more interpretable than previous approaches.

This iterative graph construction essentially means that the graph retrieval can be learnt jointly by the model, unlike the disjoint retrieval steps seen in the distractor setting models. However, just matching entities to document titles may prevent the model from detecting certain hops (where the entity may not be mentioned in a document title in the same way as in a paragraph). Furthermore, noisy sentences

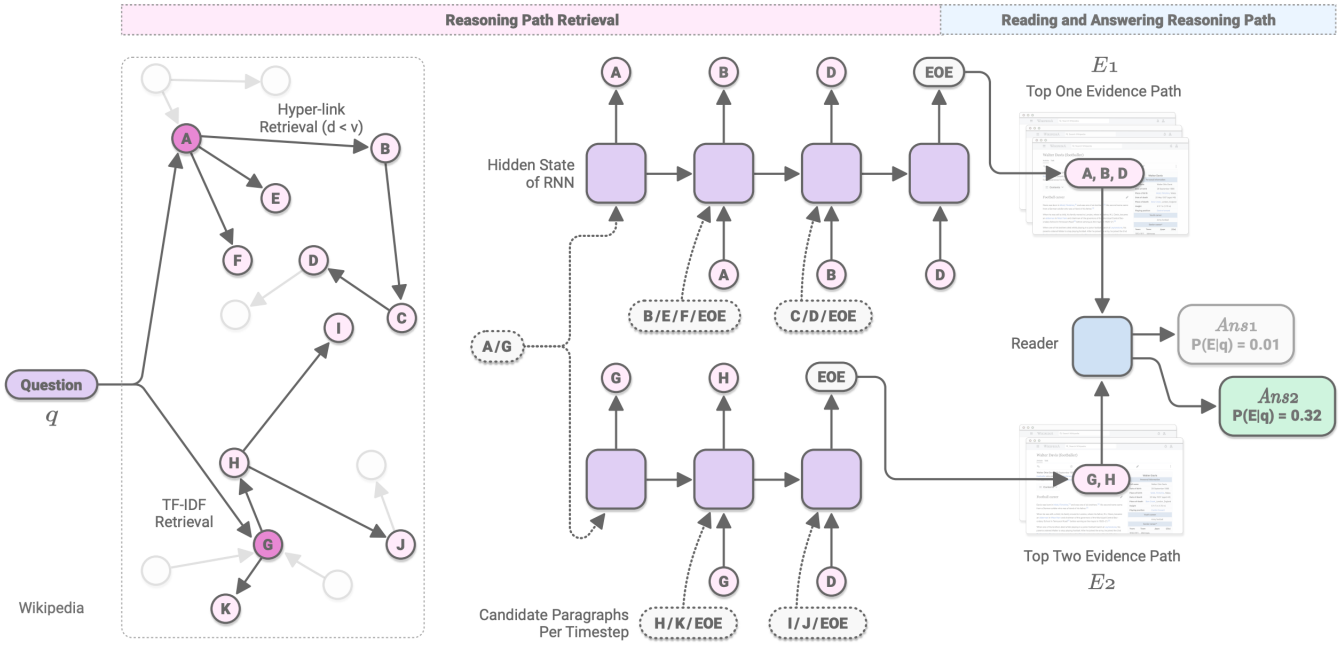


FIGURE 2.6. Diagram of recurrent retrieval model from Asai et al. (2020). The construction of two potential paths is shown, with documents represented by letters A-H.

or documents, with many potential next-hop entities, may result in more complex and thus harder to understand graphs, reducing the interpretability of this model. Finally, the authors provide no thorough examination of the interpretability of their model's cognitive graphs, instead just providing a small handful of example graphs.

In contrast to CogQA's complex entity-based approach, simpler methods for document retrieval have been shown to work just as well. Asai et al. (2020) proposed a document retrieval model for HotpotQA, utilising the hyperlinks between documents in HotpotQA (since the documents are taken from Wikipedia) to build a graph of documents. Paths through the graphs are then constructed in an iterative manner, similar to CogQA. At each step, an RNN takes in the question and a potential next-hop paragraph and predicts the probability that the given paragraph is the next hop. An 'end of path' option is also provided to allow paths of differing lengths. The most likely paths are found using beam search (keeping the top-K most likely paths at each step), with initial documents chosen by TF-IDF scores. The answer is then predicted by passing the input question and all documents in a chosen path through a BERT model, which finally predicts if the path contains the correct answer and the location of the predicted answer within the relevant documents. An overview of this architecture is given in figure 2.6.

The authors evaluated the model on HotpotQA, SQuAD, and Natural Questions (**?**) and showed improvements over CogQA and other competitive models. Furthermore, they reported improvements in determining supporting facts in HotpotQA, indicating that the model also provides some degree of interpretability. Furthermore, the authors also showed the approach works well when using an entity linker mechanism instead of hyperlinks, albeit not as well as the hyperlink-based method (likely due to the biases present with HotpotQA itself, as discussed above). However, the method does rely on document summary vectors constructed from concatenating each document with the question, which requires re-encoding every document node for each question with BERT, which is fairly costly. The authors experiment with not concatenating with the question, but find this significantly degrades results. Overall, however, this approach presents a novel and effective method for document selection which strongly inspires our own BERT-based model.

The iterative retrieval method presented in Asai et al. (2020) shares many similarities to a previous model, the multi-step retriever (Das et al., 2019). This model, although not applied to multi-hop QA, performs a similar iterative retrieval process on the SQuAD-open dataset, which shares a similar setup to HotpotQA full-wiki, apart from the fact that all questions only require one document to answer, rather than two. For this model, first, all paragraphs in the dataset are transformed into attention-based summary vectors. A given question is then similarly encoded using an attention-based summary vector. An iterative retrieve and read process is then performed: first, all paragraphs are scored by taking the dot product between the question and paragraph values. Due to the intractability of performing this operation between for millions of paragraphs, a modified nearest-neighbour algorithm is applied to find the top (closest) paragraph vector for the given question vector. Second, the chosen paragraph is passed through a regular machine reading comprehension model, such as the BiDAF or DrQA model (Seo et al., 2017; Chen et al., 2017), and an answer span is proposed along with a probability. The hidden encoded states of the question from the reading model (e.g. the question-based output from the bi-attention layer) is then used to construct a new attention-based summary vector of the question, and fed into a GRU unit along with the previous question vector to produce a new history-aware question vector. This is then used to rank the paragraphs for the next step. After some set number of steps of this iterative process, the model terminates and returns the answer span with the highest associated probability. We provide a summary diagram of this model in figure 2.7.

The model was evaluated on the TriviaQA (Joshi et al., 2017), SQuAD-open (Chen et al., 2017), Quasar-T (Dhingra et al., 2017), and SearchQA (Dunn et al., 2017) datasets, showing improvements over a

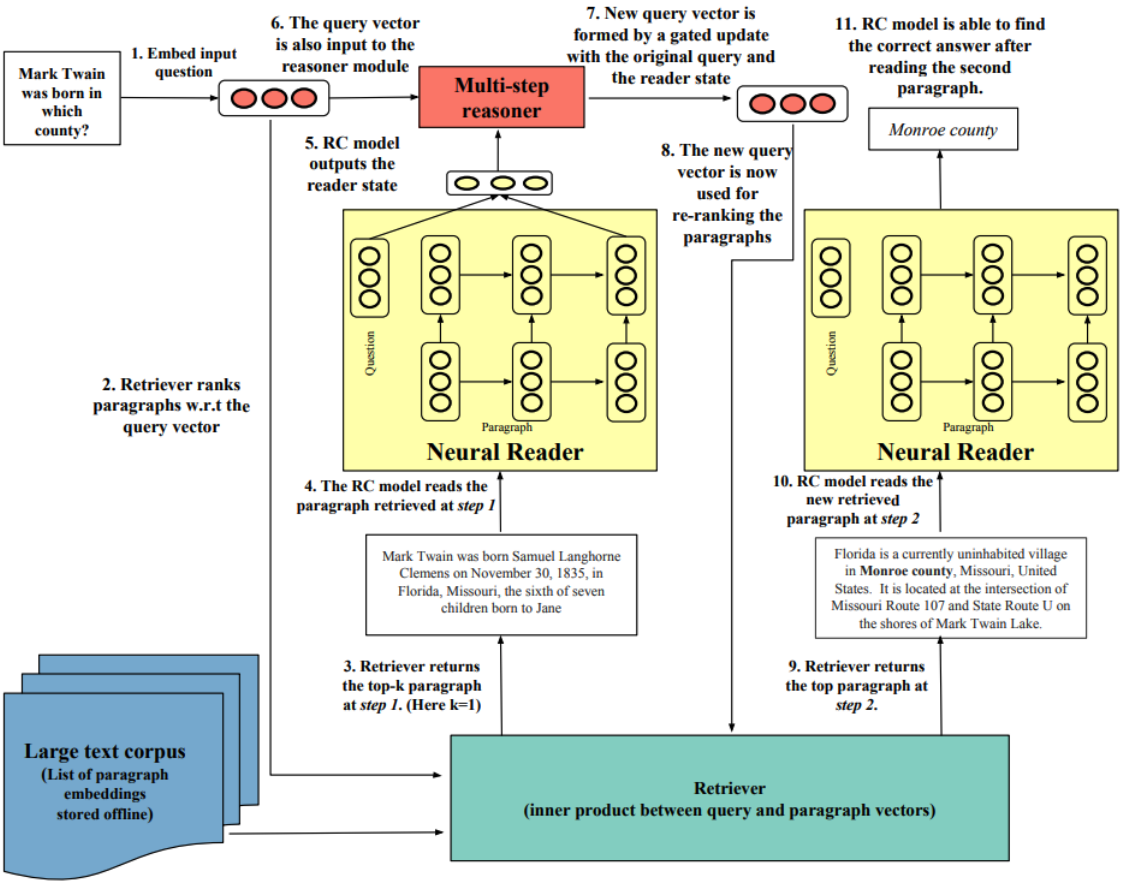


FIGURE 2.7. Summary diagram of the multi-step retreiver, from Das et al. (2019).

DrQA baseline, although it underperformed compared to a handful of strong recent models. However, further experiments showed this approach could scale up to millions of paragraphs, a size not possible for these other models, due largely their dependence on query-dependent encodings of each paragraph (thus requiring the re-encoding of every input document for every query), unlike the multi-step retriever, which used question-independent encodings. Furthermore, the multi-step approach essentially allows the model to recover from poor retrieval results by reranking documents in its next step based on information learnt from reading an incorrect document. While interesting, the model requires a complex reinforcement learning-based training method and was not applied to multi-hop settings, where following a chain of documents is required for question answering and can provide a gold training path for training the model without reinforcement learning. Other useful methods such as beam search and answer reranking were also not explored, despite their potential utility for this approach. Furthermore, integration with BERT-style models (which are common in current state-of-the-art) was not explored, although the design of the multi-step retriever could easily be adapted to utilise BERT. Thus this is a

TABLE 2.2. Summary of models discussed in section 2.5. N/A indicates the interpretability of the model was not discussed in that paper.

Paper	Model	Dataset(s)	Interpretability?
Qiu et al. (2019)	DFGN	HotpotQA (distractor)	Attention weights
Tu et al. (2020)	SAE	HotpotQA (distractor)	Attention weights
Fang et al. (2020)	HGN	HotpotQA (distractor)	N/A
Ding et al. (2019)	CogQA	HotpotQA (full-wiki)	Reasoning path
Asai et al. (2020)	Recurrent Retriever	HotpotQA (full-wiki), SquAD, Natural Questions	Reasoning path
Das et al. (2019)	Multi-step Retriever	TriviaQA, Quasar-T, SearchQA, SQuAD-open	Reasoning path

promising but under-explored model that utilises a recurrent retrieval process to great effect on non-multihop datasets, integrating the retrieval and reading components more tightly than Asai et al. (2020). Augmenting this model with more modern open-domain QA components and the improvements made in Asai et al. (2020) is an interesting and promising path of research, which we explore in this work.

Thus, we have shown how iterative retrieval methods are a promising paradigm for QA, both for multi-hop and non-multi-hop datasets. Furthermore, by examining the steps taken by these models, their reasoning processes can be made somewhat clear, as one can trace their reasoning path through a set of documents. However, interpretability across both settings is somewhat lacking due to a large reliance on BERT for answer prediction. Furthermore, while Das et al. (2019) and Ding et al. (2019) tightly integrate their retrieval and answer prediction mechanisms, Asai et al. (2020) does not, utilising separate reading and retrieval modules. Thus, exploring tighter integration of these recurrent retrieval mechanisms with the document reading and reasoning process is clearly an interesting and promising line of research.

We summarise the papers mentioned in this section in table 2.2.

2.6 MAC Networks and Text-based Reasoning

2.6.1 MAC Networks

Compositional attention networks (CANs, or more commonly referred to as MAC networks) (Hudson and Manning, 2018) are a novel network design for the visual question answering task, where a model is tasked with answering questions about an image. The architecture is primarily based around a recurrent cell: the memory, attention, composition (MAC) cell, which is designed to model basic reasoning components required for compositional reasoning. As seen in figure 4.2, each cell contains three units,

one for reading the question, one for reading the image, and then one for integrating information from the image into a memory state based on the information from the question. All cells use a mixture of attention and linear layers to perform their specific tasks. Each cell also takes in and outputs a control state and a memory state in a recurrent manner. The control state is only manipulated by the control unit and represents information extracted by the model from the question. This is passed to the read and write units to then guide both the extraction of information from the image by the read unit and the integration of the read information into the memory state by the write unit. The memory state thus represents useful information extracted from the knowledge base (i.e. image), which is used to perform a final answer prediction. We provide a more in-depth description of the MAC cell in chapter 4.

MAC networks carry several benefits over other approaches to VQA, achieving state-of-the-art accuracy on the CLEVR dataset (Johnson et al., 2017) with less compute than other approaches, and exhibits better generalising ability, achieving high accuracy even with small subsets of training data. Finally, the reasoning steps taken by the network can be visualised by examining the attention maps produced by each cell over the input image and question, meaning the design is reasonably interpretable. Thus, MAC networks are clearly promising candidates for applying to interpretable multi-step reasoning.

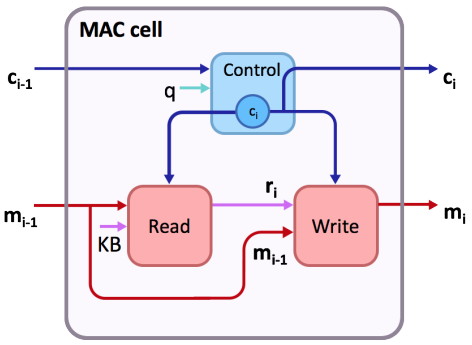


FIGURE 2.8. Diagram of a MAC cell from Hudson and Manning (2018). c_i and m_i represent the control and memory states at step i, while q and KB represent the encoded question and knowledge base. r_i represents the information extracted by the read unit at step i.

2.6.2 Applying MAC Networks to Text

Despite the success of the MAC network, relatively few papers have examined applying MAC cells to text-based QA tasks. Below we describe work that applies MAC-inspired designs to QA tasks where all input is purely text.

2.6.2.1 Multiple-choice QA with MAC networks

Yu et al. (2019) proposed a model close to the original MAC design and applied it to multi-step ('inferential') multiple-choice question answering, where a list of possible answers is given alongside a document and question. The proposed model mainly utilises a novel cell largely similar in design to the MAC cell, the largest difference being the use of a dynamic number of cells rather than a fixed number.

The authors evaluated against a variety of multiple-choice QA datasets and found that the MAC-inspired design outperformed the state-of-the-art while retaining some of its interpretable qualities, suggesting that the MAC cell is a good candidate for text-based QA. However, they require an overly complex reinforcement learning-based training method to train the dynamic number of cells, rather than utilise the existing mechanisms in the MAC network which allow for arbitrary-length reasoning without complex training. Furthermore, the potential answer options are tightly integrated into their cell design, providing closer guidance to the reasoning process than is possible in span-based QA tasks, where there is no list of potential answers for the model to refer to. Finally, the authors do not make use of pretrained models such as BERT, variants of which have since out-performed this model on multiple-choice QA datasets (Pan et al., 2019; Jin et al., 2020; Wang et al., 2019a).

More recently, Le Berre and Langlais (2020) also apply MAC networks to multiple-choice QA, focusing on the ARC (Clark et al., 2018) and OpenBook QA (Mihaylov et al., 2018) datasets, which focus more on commonsense questions. They experiment with integrating a MAC network with an existing BERT model for these datasets, and find it provides minimal benefits in very particular setups. This indicates that while the MAC network may be useful, it requires either further adaption or a different overall design to provide substantial performance gains to a BERT-based model. Sinha et al. (2019) also apply MAC networks to a novel multi-choice QA dataset called CLUTRR, which tests a model's ability to learn and apply graph-based logical rules from natural language text. They show that a MAC network is competitive at detecting supporting facts and more robust to noise than other models, although less performant than a basic BERT model. However, they do not explore augmenting the MAC network with BERT or in other ways, meaning there are many further ways to improve on their usage of MAC networks.

2.6.2.2 MAC for expert-finding

Fu et al. (2020) introduced a modified MAC network, called the 'recurrent memory reasoning network', in order to tackle the task of identifying users best suited to answering a given question. For this task, a network is provided with a question, a candidate expert, and a list of previous answers given by the candidate expert. The network design is largely similar to the original MAC cell design, its largest difference being that the authors add a document selection mechanism into the read unit, allowing each MAC cell to select a set of k answers given by a given expert, and use those as the knowledge base for that cell. This allows the MAC network to shrink the size of the input to its regular read unit, and the

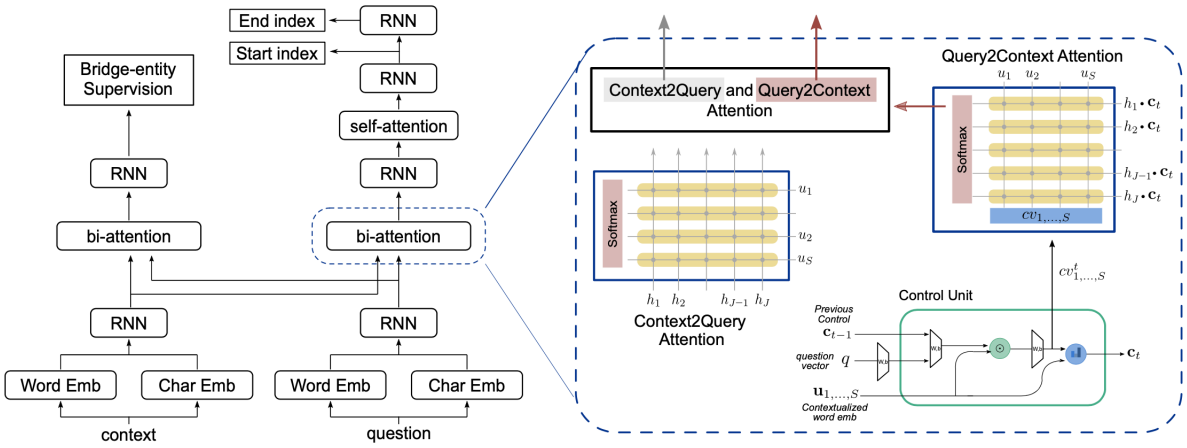


FIGURE 2.9. Diagram from (Jiang and Bansal, 2019a), showing their augmented bi-attention flow model. The output of the control unit is used to bias the Query2Context attention. Both attention distributions are otherwise calculated as in (Seo et al., 2017).

authors report this mechanism results in improved performance for the expert-finding task. Therefore, this is another clear example of the utility of the MAC for different text-based reasoning problems, and proposes a useful method for allowing the MAC network to process large numbers of documents (when only a subset are relevant to the task). However, the task of expert-finding is somewhat simpler than QA, as it does not require token-level reasoning nor more complex reasoning - rather, the model simply needs to accumulate evidence that a user contains enough expertise for a given question. Furthermore, the authors do not examine using BERT, which again may provide large performance gains.

2.6.2.3 Control-Augmented DocQA

Jiang and Bansal (2019a) critiqued and improved the HotpotQA distractor setting dataset by showing that many questions in the dataset can be solved unintentionally in a single hop and constructing a more challenging dataset by using adversarial methods. In particular, they compared the performance of a DocQA model (Clark and Gardner, 2018) with a DocQA model augmented with a control unit (from the MAC cell design) on both the original HotpotQA and their own adversarially-generated dataset. The architecture for the augmented model can be seen in figure 2.9, in which the output of the control unit (an attention distribution over the question words) is used to bias the bi-attention layer. The control unit allows the network to focus on sub-parts of the question, which it is explicitly trained to do by training the model to find the entity linking the two supporting documents (called the 'bridge entity'), which is called 'bridge entity supervision'.

The addition of the control unit improved performance across all tested datasets (when trained to predict supporting facts as well as answers) and is more robust to adversarial data. Furthermore, the 2-hop model with control unit removed performed reasonably worse than the 2-hop model with the control unit, suggesting that the addition of the control unit improves the model's ability to perform multi-step reasoning. Hence, this paper shows that MAC cells are a promising candidate for further research into multi-step reasoning models and that augmenting existing models with ideas or components from MAC cells can improve their performance. However, despite the success of their control-augmented model, the authors do not test any models closer to the original MAC architecture on their datasets and do not test more than two 'hops' in their model. Thus, MAC cells are clearly promising candidates for good performance on the HotpotQA dataset.

2.6.2.4 Neural Module Networks (NMNs) for Text-based Reasoning

Closely related to MAC networks are neural module networks (Andreas et al., 2016), which dynamically select neural modules (i.e. sub-neural networks) to perform different reasoning steps before reaching an answer. In doing so, neural module networks are interpretable, as one can see the modules chosen, and flexible, as one can simply design new modules for new types of reasoning. While initially only applied to visual question answering, recent work has examined adapting these to complex text-based reasoning, primarily examining multi-hop reasoning (Jiang and Bansal, 2019b) and discrete reasoning (Gupta et al., 2020). Jiang and Bansal (2019b) adapted the neural module network to the HotpotQA dataset, utilising not only a control unit similar to the MAC network, but also a memory state and iterative steps of attention to perform the multi-hop reasoning required by the dataset. While the model can be trained end-to-end (i.e. just on the question and answer pairs), they find that additional supervision on the network is required to guide its reasoning process, resulting in higher scores on the HotpotQA dataset. They also perform a short examination of integrating BERT-based embeddings into their model, finding that it performs above a vanilla BERT model. However, they use a fairly naive approach for document selection with their BERT model, when recent work has shown improved document selection techniques can greatly improve the performance of BERT (Tu et al., 2020). Furthermore, the authors did not examine the integration of supporting facts into their model, which would potentially provide better supervision for the reasoning process than their naive heuristic for finding a bridge entity. Gupta et al. (2020) also adapt the neural module network to QA, but focus on discrete reasoning rather than multi-hop reasoning, and similarly find that they can achieve competitive performance only when utilising auxiliary supervision to guide its reasoning process.

2.6.2.5 Summary

Therefore, we can see that MAC networks show various improvements over existing models: they are better able to perform multi-step reasoning than existing popular QA models across various datasets and provide in-built interpretability via their attention maps. Whilst several works have adapted MAC networks to multiple-choice QA (Yu et al., 2019; Le Berre and Langlais, 2020), no work has yet closely examining adapting the MAC architecture to span-based QA. In addition, the incorporation of MAC-like elements for span-based multi-hop QA (Jiang and Bansal, 2019a) and the success of neural module networks on span-based QA (Jiang and Bansal, 2019b; Gupta et al., 2020) suggests that the MAC design has promise for these tasks. Furthermore, little work examines incorporating BERT-based embeddings into these architectures, with existing work focussing on the use of GloVe-based embeddings (Hudson and Manning, 2018; Yu et al., 2019; Jiang and Bansal, 2019a,b). Thus, we can see that the application of MAC networks for span-based QA is an under-explored but promising line of research, with no existing work examining a full adaption the MAC model to span-based QA. We address this gap in this work, examining how we can keep the core ideas of the MAC cell intact while adapting it to span-based QA.

We summarise the papers presented in this section in table 2.3.

Paper	**Model**	**Task**	**Dataset**
Hudson and Manning (2018)	MAC	Visual multi-choice QA*	CLEVR, CLEVR-humans
Yu et al. (2019)	Micro-infer Cells	Text-only multi-choice QA	MultiRC, RACE, MCTest
Le Berre and Langlais (2020)	BERT + MAC	Commonsense multi-choice QA	ARC, OpenBook QA
Jiang and Bansal (2019a)	Control + DocQA	Text-only multi-hop QA	HotpotQA (original and adversarial)
Jiang and Bansal (2019b)	Hotpot-NMN	Text-only multi-hop QA	HotpotQA (original and adversarial)
Gupta et al. (2020)	Text-NMN	Text-only complex QA	DROP
Sinha et al. (2019)	MAC	Inductive reasoning	CLUTRR (new dataset)

TABLE 2.3. Summary of models presented in section 2.6. * The CLEVR dataset has a closed answer set, and so the MAC treates it like a multiple-choice answer dataset, where the model simply chooses the most likely answer from a large list of possible answers.

2.7 Conclusion

While there has been much research into complex and multi-hop question answering, current approaches generally revolve around the use of large pretrained language models and complex graph-based designs, despite such complex graph approaches being potentially unnecessarily complex. The reliance on pretrained language models also provides little in the way of interpretability, despite that being a primary task provided by the HotpotQA dataset. Furthermore, several models keep the reasoning process for supporting facts and answer prediction distinct (Tu et al., 2020; Asai et al., 2020), going against the purpose of the supporting facts to provide an insight into the reasoning process of the model. In contrast, MAC-style models provide a method to 'peer inside' their reasoning process, through the use of attention maps. In addition, the multi-step design of MAC models seems naturally suited to the task of multi-hop reasoning, being similar to the approach explored in Qiu et al. (2019).

However, no work has yet examined in-depth the application of MAC networks to span-based multi-hop reasoning, despite their success not just in VQA, but also in multiple-choice question answering. Whilst adapting modular networks to span-based question answering is difficult, as evidenced by recent work on neural module networks, it can still lead to competitive results, potentially resulting in models that are simultaneously more powerful and interpretable than existing models for multi-step reasoning. As such, MAC cells are clearly promising candidates for further research in relation to multi-hop QA, which we will explore in the following chapters.

CHAPTER 3

Evaluation

In this chapter, we provide a detailed description of the datasets used for the evaluation of our work, focusing on HotpotQA, a popular multi-hop reading comprehension dataset. We also list the various baseline and state-of-the-art models in multi-hop reasoning used to compare our model against. Finally, we describe the metrics and methods we will use to evaluate models on these datasets.

3.1 Datasets

3.1.1 HotpotQA

We primarily focus our evaluation on the HotpotQA dataset (Yang et al., 2018), a popular multi-hop reading comprehension dataset. This dataset has inspired a flurry of recent research, with recent methods substantially improving on the baseline models provided along with the dataset (Fang et al., 2020; Asai et al., 2020). However, a substantial gap still exists between the current state-of-the-art and human performance, indicating there are further possible gains in performance. The dataset consists of human-written questions that require reading across multiple documents. The documents are first paragraphs from a Wikipedia dump. In addition to answers, HotpotQA also provides annotations of 'supporting facts': labelled sentences indicating if a given sentence from a document supports the answer. These supporting facts thus provide a way to evaluate the interpretability of a model, and encourage more interpretable model designs (albeit in a very specific way). As such, HotpotQA provides an excellent testbed for building novel interpretable models for multi-hop reading comprehension.

The HotpotQA dataset contains two setups for models: the *distractor* setting, in which each question is paired with a set of ten documents (two of which are necessary to read to find the correct answer), and the *full-wiki* setting, in which documents are not provided, and models have to find relevant documents from a set Wikipedia dump themselves. As there are over 5 million documents in the provided Wikipedia

Split	# Examples
train	90,447
dev	7,405
test-distractor	7,405
test-fullwiki	7,405

TABLE 3.1. Number of examples in each split of the HotpotQA dataset.

dump (Yang et al., 2018), this is a non-trivial task. This provides two clear paths for research: improving the reading ability of models given relevant documents (tested in both setups), and improving the ability of models to retrieve relevant documents (tested in the full-wiki setup). While both setups have their own test set, they share train and development sets. Note that these test sets are completely private and can only be evaluated via submission to the official HotpotQA website. We give summary statistics of the overall dataset in table 3.1.

Across both setups, there are two types of questions present in HotpotQA: *bridge* questions, which require finding out the identity of or facts about some entity linking two parts of the question (the 'bridge entity'), and *comparison* questions, which simply require comparing two entities in some way. An example of a bridge question is *'2014 S/S is the debut album of a South Korean boy group that was formed by who?'*. To answer this question, one must first find out the name of the boy group being referred to before they can determine who they were formed by. Therefore, the name of the boy group ('Winner') is the bridge entity in this question. An example of a comparison question would be *'Did LostAlone and Guster have the same number of members?'*. To answer this question, one must first find out the number of members each band has before comparing the two. Comparison questions can further be divided into *polar* comparison questions, which just require a yes/no response (as the previous example), and *non-polar* comparison questions, which require answering with an entity name, for example: *'Who is older, Annie Morton or Terry Richardson?'*. All non-polar questions (bridge and comparison) can be answered with a text span from the input documents. Models built for HotpotQA thus generally adopt a two-stage answer prediction process: first, they predict if the answer to a given question is 'yes', 'no', or a span in the text. If it is a span, they then predict the start and end location of the text span that serves as the answer. We provide summary statistics of each question type in table 3.2.

We can also split up the HotpotQA dataset by answer type. While answer type labels were not explicitly collected by the authors, we hand-labelled 300 randomly-chosen examples from the development set in order to gain a greater understanding of the types of reasoning required. We provide the summary statistics of each type in table 3.3, and the descriptions of each label can be found below:

Split	Question Type	# Examples
train	bridge	72,991 (80.7%)
	polar comparison	5,481 (6.1%)
	non-polar comparison	11,975 (13.2%)
dev	bridge	5,918 (79.9%)
	polar comparison	458 (6.2%)
	non-polar comparison	1,029 (13.9%)

TABLE 3.2. Number of different question types in train and dev splits. Test counts are unknown due to test set being kept private.

- **Yes/No**: Answer is yes/no
- **Adjective**: Answer is an adjective
- **Artwork**: Answer is the name of an artwork, including visual art, songs, tv shows, etc.
- **Event**: Answer is the name of an event (for example, a race or sporting event).
- **Group**: Answer is the name of a company, team, or organisation.
- **Location**: Answer is a place name.
- **Person**: Answer is a person name.
- **Number**: Answer is a number.
- **Date**: Answer is a date.
- **Common noun**: Answer is a general common noun that doesn't fit into other categories (for example, the name of a profession).
- **Proper noun**: Answer is a proper noun that doesn't fit into other categories (for example, the name of a product).
- **Mislabel**: Answer is clearly incorrect or nonsensical.

3.1.2 Adversarial HotpotQA

While HotpotQA has been a popular and useful dataset for studying multi-hop question answering, several papers have called its multi-hop nature into question in the distractor setting, showing that various models are able to exploit keywords in questions and skip the intended reasoning paths to be able to answer questions without drawing upon multiple documents (Jiang and Bansal, 2019a; Min et al., 2019). To remedy this, Jiang and Bansal (2019a) propose an adversarially-generated dataset that provides stronger distractors than regular HotpotQA. This dataset is constructed by finding the document containing the answer, and replacing mentions of the answer in it with slightly modified answer phrases (e.g. if the answer was 'World's best goalkeeper', a perturbed version would be 'World's best defender').

Answer Type	# Examples
Person	73 (24.3%)
Location	54 (18.0%)
Group	27 (9.0%)
Date	27 (9.0%)
Proper noun	27 (9.0%)
Number	23 (7.7%)
Yes/No	21 (7.0%)
Artwork	16 (5.3%)
Common noun	15 (5.0%)
Adjective	9 (3.0%)
Event	6 (2.0%)
Mislabel	2 (0.7%)

TABLE 3.3. Number of different answer types from a random sample of 300 questions from development set. See section 3.1.1 for details.

In addition, the bridge entity used to derive the original answer is replaced with a randomly sampled entity, and all mentions of the bridge entity in the perturbed document are replaced with the sampled one. This means that the perturbed document still satisfies the shortcut, but does not contain the correct answer or the correct bridge entity. Four perturbed documents are constructed and then inserted into the dataset by replacing 4 random distractor documents. Thus, by providing these perturbed documents as strong distractors, we can better evaluate the multi-hop reasoning ability of our models. We follow Jiang and Bansal (2019a) in evaluating on this adversarial dataset: we randomly sample 40% of the adversarially-generated data and mix it into the regular HotpotQA training data, and then evaluate on a fully adversarially-constructed development set. During testing, we also found that the code provided by the authors[1] marked their augmented distractor documents in their title[2], allowing models trained on the adversarial data to easily identify the new distractors and ignore them. To remedy this, we altered their code to remove this mark, replacing it with a title that reflects the changes made in the document itself. We apply this change to the generation of the adversarial training and development sets, and use these modified sets for our own evaluation.

3.1.3 Single-Hop Reading Comprehension

In order to test our models' generalising ability, we measure its performance on the popular non-multi-hop reading comprehension dataset SQuAD, using both versions 1.1 (Rajpurkar et al., 2016) and 2.0

[1]Available at https://github.com/jiangycTarheel/Adversarial-MultiHopQA

[2]Specifically, the title of generated documents would always be 'added', rather than an actual title.

(Rajpurkar et al., 2018). These datasets share the same answer F1 and EM metrics as HotpotQA (explained in section 3.3) but ask simpler questions that do not require multiple documents to answer. SQuAD 1.1 contains comprehension questions on small input documents (usually one paragraph from a Wikipedia page), focusing purely on a model's ability to understand short paragraphs of text. In addition, SQuAD 2.0 adds questions with no answers into the SQuAD 1.1 dataset, requiring models to not only locate answers, but also determine if there is enough evidence in a given text in order to answer a question. We utilise the no answer module from the DocQA model for no-answer prediction in SQuAD 2.0.

3.2 Baselines and State-of-the-art

In this work, we compare against several other models, including the original strong baseline for HotpotQA, and more recent work on the dataset. We provide a list of models we compare against below:

- **HotpotQA Baseline (DocQA)**: This is the baseline model described in Yang et al. (2018), and is an adapted version of the popular DocQA model (Clark and Gardner, 2018) for the HotpotQA dataset. This model utilises three key features from previous state-of-the-art models: character-based embeddings combined with GloVe embeddings, a bi-attention layer, and a self-attention layer. While this model does not use BERT-based embeddings (a core component of current state-of-the-art approaches to question answering), it still remains a strong baseline for non-BERT based methods.
- **Control + DocQA**: This is the baseline model augmented with a control unit introduced by Jiang and Bansal (2019a). This utilises the control-augmented bi-attention layer used in our model by adding a control unit to the DocQA model (although not utilising the other aspects of the MAC cell design).
- **Hotpot-NMN**: This is a neural module network adapted to the HotpotQA dataset, introduced in Jiang and Bansal (2019b). The network contains 3 modules which are dynamically assembled for each input question, allowing custom logic for different question types. The overall network design contains elements similar to the MAC network, including the control unit, and so is a good example of a GloVe-based modular network that performs above the baseline model.
- **BERT**: We use a basic BERT model (Devlin et al., 2019) with the supporting fact prediction design used by the baseline model (an RNN on top of the BERT model for constructing

sentence embeddings) as a baseline for determining if our designs provide any performance improvements over BERT.

- **Select, Answer, and Explain (SAE) Model**: This model was introduced in Tu et al. (2020) for the distractor setting, and utilises a complex document selection process involving a BERT model to determine the correct two documents required to answer the question. Once the documents have been selected, they are fed into a BERT model to predict the answer location. In order to predict answer type ('yes', 'no', or 'span') and supporting facts, sentence embeddings are constructed from the BERT output and a graph is constructed based on basic entity linking between sentences. This graph is then passed through a graph neural network to make supporting fact and answer type predictions. While this graph-based step is complex, the main answering component of this model is simple, just using BERT to directly predict answer locations from a narrowed-down set of documents. As such, this model is a good example of the upper limits of performance when using plain BERT-based models.
- **Hierachical Graph Network (HGN)**: This model was introduced in Fang et al. (2020), and is currently the highest-performing published model on the HotpotQA distractor setting[3]. This model utilises a hierarchical graph network consisting of entity, sentence and document nodes to both augment the predictions from a BERT-based model and to make supporting fact predictions. In addition, it utilises the hyperlinks originally present in the Wikipedia documents to link relevant documents in its graph.
- **Recurrent Retriever (RR)**: This model, introduced in Asai et al. (2020), is a highly competitive model that establishes a new method for document retrieval in the HotpotQA full-wiki setting. This model utilises a recurrent unit to perform a graph-based search over hyperlinked documents as part of its document retrieval process, before using a BERT model to rank the best paths found by the recurrent unit and predict the final answer. As the strongest published model with a recurrent retrieval process, this model is a strong baseline for comparisons against our own recurrent retrieval model.

For comparisons, we restrict comparisons of GloVe-based models to only other GloVe-based models (DocQA, Control+DocQA, Hotpot-NMN), to ensure that comparisons are fair, since BERT along performs incredibly strongly on HotpotQA, and it is non-trivial to integrate it into existing GloVe-based approaches. Integrating BERT into our approach is studied in detail in chapters 5 and 7, in which we do compare against models that utilise BERT (SAE, HGN, RR).

[3] At time of writing, November 18, 2020.

3.3 Metrics

Both HotpotQA and adversarial HotpotQA utilise the same set of metrics for performance evaluation: F1-score and exact match score. Three sets of these metrics are calculated: one set each for answer-only evaluation, supporting fact-only evaluation, and joint answer and supporting fact evaluation. The answer F1-score is calculated by first normalising both the ground truth and predicted answers by case-folding (changing everything to lowercase), normalising whitespace to be single spaces only, removing punctuation, and removing 'a', 'an', and 'the'. After this, tokens are determined by splitting the answers on whitespace, and the final score is calculated as:

$$P_a = \frac{\text{num. tokens same between true and predicted answers}}{\text{total tokens in predicted answer}}$$
$$R_a = \frac{\text{num. tokens same between true and predicted answers}}{\text{total tokens in ground truth answer}}$$
$$\text{F1}_a = \frac{2P_aR_a}{P_a + R_a}$$

The answer exact match score is calculated by following the same normalisation procedure as for the F1-score, but then just checking if the two normalised strings are identical. If so, the score is 1, else it is 0.

The supporting fact metrics are determined in the ordinary manner. First, the number of true positives (correctly identified supporting sentences), false positives (incorrectly predicted supporting sentences), and false negatives (non-predicted supporting sentences) are counted. The F1 and exact match scores are then calculated as:

$$P_{sp} = \frac{tp}{tp + fp}$$
$$R_{sp} = \frac{tp}{tp + fn}$$
$$\text{F1}_{sp} = \frac{2P_{sp}R_{sp}}{P_{sp} + R_{sp}}$$
$$\text{EM}_{sp} = \begin{cases} 1, & \text{if } fn + fp = 0 \\ 0, & \text{otherwise} \end{cases}$$

Where tp, fp, and fn represent the number of true positives, false positives, and false negatives respectively.

Finally, the joint F1 score is calculated as:

$$P_j = P_a P_{sp}$$

$$R_j = R_a R_{sp}$$

$$\text{F1}_j = \frac{2P_j R_j}{P_j + R_j}$$

The joint exact match score is simply 1 if both the answer and supporting fact exact match scores are 1, and 0 otherwise. We report the average of the metrics described above, calculated across the entirety of the dataset being evaluated on.

3.4 Evaluation Methods

We evaluate our models both quantitatively and qualitatively. For both models, we perform the following quantitative evaluations, which are standard in deep learning-based NLP work, and important for distinguishing where our models improve on existing work (Lipton and Steinhardt, 2019).

- **Performance**: We evaluate the overall performance of our final models on various datasets, including HotpotQA and adversarial HotpotQA. When testing on adversarial HotpotQA, we follow Jiang and Bansal (2019a) in reporting performance on both HotpotQA variants when trained on adversarial or non-adversarial HotpotQA. We note that we found a small issue with the adversarial dataset generation that we fixed (detailed above), and resulted in quite different performance to that reported in Jiang and Bansal (2019a). As such, we limit our adversarial evaluation comparisons to comparisons between the baseline and our models on the fixed adversarial dataset (as these are the models we have access to training code for).
- **Ablations**: We remove or alter various components of our models and then evaluate performance on HotpotQA to determine which are the most effective and their individual contributions to our overall performance.

- **Parameter tuning**: We evaluate our model on HotpotQA while changing various hyperparameters, including the learning rate, to investigate their effect on the performance of our models.

We also provide the following qualitative evaluations of our models, investigating the internal workings and general behaviours of our models.

- **Attention maps**: We select some samples from the development set and visualise the attention inside our models. This provides us insight into the interpretability of our models as well as some degree of their inner workings. This evaluation is similar to evaluations provided in Hudson and Manning (2018), Qiu et al. (2019), and Tu et al. (2020).
- **Question and answer type breakdown**: We breakdown each models' performance using the questions labelled by answer type as described in section 3.1.1, the question types provided in the dataset itself (bridge, polar comparison, and non-polar comparison), and by the combined length of the input documents. This provides us insight into which sort of questions our model performs well on and which sorts it performs poorly on. This is an extension on the evaluation provided by existing work, which largely only breaks down performance only by question type (Qiu et al., 2019; Jiang and Bansal, 2019b; Tu et al., 2020).
- **Error analysis**: We examine a randomly-chosen sample subset of question-answer pairs our models get completely wrong (0 F1) and label them with the reason for their low performance. This provides us with insight into the failures of our model and potential areas for future work. This is loosely based on the analysis performed in Fang et al. (2020).

CHAPTER 4

GloVe-based Model

In this chapter, we thoroughly describe the design of our GloVe-based model, the first model proposed in chapter 1. This model is designed to take in a question and list of documents (all represented via raw text), and output an answer to the given question, which is either a span of text from one of the provided documents or a simple 'yes' or 'no'. It consists of three general components:

- **Encoding unit**: We employ a model to encode documents and questions, mapping sub-word units to contextually-aware vector representations. In this model, this is done using GloVe vectors, character embeddings and bidirectional GRU layers (Cho et al., 2014). We also construct a dense vector representation of the question.
- **Recurrent Memory, Attention, Composition (MAC) Cell**: We feed our vector representations in a recurrent MAC cell, which performs multi-step reasoning. In our baseline-based model, the cell simply uses a modified bi-attention layer to calculate interactions between the question and the input documents.
- **Output unit**: Finally, we use the outputs from the final MAC cell to predict the answer to a given question, along with supporting facts. This is done using a final self-attention layer and stack of GRU layers.

A diagram of this model is given in figure 4.1.

In the below descriptions, we use $W^{d \times b}$ to denote weight matrices of dimension $d \times b$, where d and b are integers. We use b to indicate a weight matrix used for bias. Thus, W and b represent trainable parameters in our model. Although we use mini-batches during training, we leave out the batch dimension in our description.

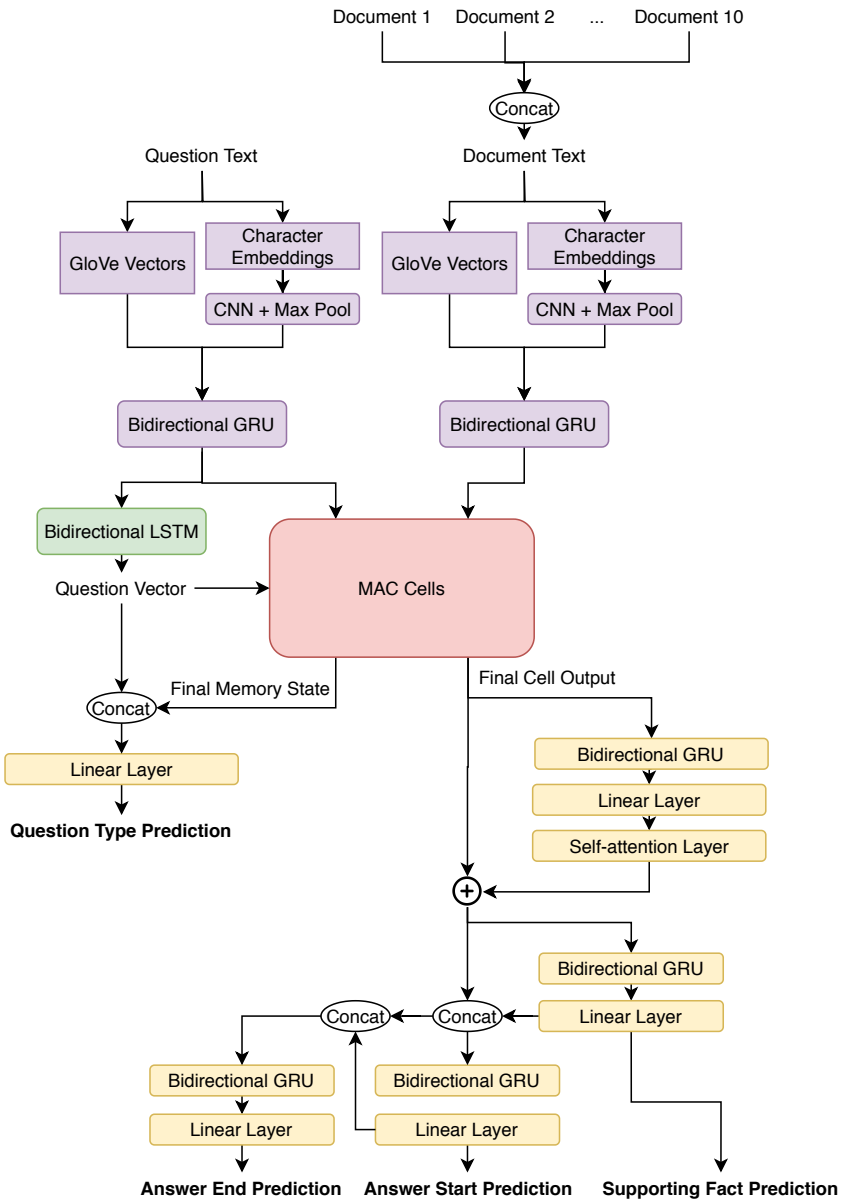


FIGURE 4.1. High-level architecture diagram of our GloVe-based model.

4.1 Text Encoding

Before we can perform any reasoning, we first need to convert our text into a form that our model can utilise - word embeddings. These are dense vector representations of words, which contain various semantic and syntactic information about the given word or sub-word unit.

4.1.1 Text Preparation

We first concatenate the document text into one long string following the pattern '<t> Title 1 </t> Text 1 <t> Title 2 </t> Text 2...', where 'title 1' refers to the title of the first document, and 'text 1' refers to the text of the first document. For the rest of the model, the multiple documents are simply treated together in this concatenated form. This allows multi-document reasoning to be approached identically to single document reasoning at the cost of larger inputs. We then tokenize all our input text (document and question) using the NLTK tokenizer (Bird and Loper, 2004), which turns our input data into a list of words. We then construct a list of all tokenized words and characters that appear in our training text and assign vectors for each word and character. For words, these vectors are assigned using pretrained GloVe vectors (Pennington et al., 2014), while for characters, they are simply randomly generated. Following the baseline model, we limit the size of our lists (our vocabulary size) to 2,200,000 words, and 94 characters (with higher-frequency words and characters taking priority over lower-frequency ones). This captures the vast majority of the characters and words present in the training text, with any leftovers mapped to a special 'UNK' vector, which represents an unknown word. By using this special token and character-level embeddings, the model can better reason about previously unseen words at test time. In addition, these initialised vectors are trained during training, allowing our model to fine-tune these word representations to best suit its task. Each GloVe vector has dimensionality 300 and each character vector has dimensionality 8.

4.1.2 Text Encoding

With our text now represented by vectors, we add some additional processing to enrich the information stored in each vector. First, we apply one-dimensional convolution to the character embeddings of each word, and then utilise max-pooling to construct a single character-aware vector for each word. Formally, if we have a sequence of n character embeddings $[x_0, x_1, ..., x_n]$ making up a single word and a convolution kernel size of $2k + 1$, then the character-aware word vector is constructed by:

$$x'_i = f(W^{8\cdot(2k+1)\times 100} \cdot x_{[i-k:i+k]} + b), \text{where } i = 0 \dots n \tag{4.1}$$

$$c = max([x'_0, x'_1, ..., x'_n]) \tag{4.2}$$

Where c represents our character-aware word vector. Following the baseline model, we set $k = 2$. We then concatenate c with the word embedding vector, and pass this through a single gated recurrent unit (GRU) layer. The GRU is a recurrent cell that has similar performance, but lower computational cost, than the popular long short-term memory (LSTM) cell (Hochreiter and Schmidhuber, 1997). It processes a sequence step-by-step, utilising a hidden state as memory to identify dependencies between items in the sequence. At each step, transformed representations of the item of the sequence input at that step are produced, allowing us to enrich our word vectors with knowledge of their textual context (including surrounding words, syntax, and so on). Formally, at timestep t, the GRU takes in the t^{th} item of a sequence x_t, and previous hidden state h_{t-1}, and outputs a single hidden state, which can be used as the transformed representation of x_t, and is input to the next step (assuming a hidden and input dimension of d):

$$z_t = \sigma_g(W^{d\times d}x_t + W^{d\times d}h_{t-1} + b) \tag{4.3}$$

$$r_t = \sigma_g(W^{d\times d}x_t + W^{d\times d}h_{t-1} + b) \tag{4.4}$$

$$\hat{h}_t = \tanh(W^{d\times d}x_t + W^{d\times d}(r_t \odot h_{t-1}) + b) \tag{4.5}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \tag{4.6}$$

We employ two GRU layers, one going forwards through the text sequence and one going backwards, to allow our word vectors to be aware of their prior and future context. Our final encoded word vectors are then formed by concatenating the outputs for the same word from the two directions. Formally, if we have a sequence of n word vectors $[x_0, x_1, ..., x_n]$, then our final representation of the word at timestep t, x'_t is given by:

$$h^f_t = \text{GRU}_f(x_t, h^f_{t-1}) \tag{4.7}$$

$$h^b_t = \text{GRU}_b(x_t, h^b_{t+1}) \tag{4.8}$$

$$x'_t = [h^f_t; h^b_t] \tag{4.9}$$

We use 'GRU' to represent the set of equations 4.3-4.6. As mentioned above, this encoding step is applied in the same manner to the concatenated document text and question text, but using separately-trained layers (i.e. the equations are the same, but the weights used are distinct). This means that the document representations are not aware of the question representations, and vice-versa. Instead, interactions between the context and documents are handled by our MAC cells, described below. By utilising the above process, we have constructed representations of each tokenized word that are not only possible to train with the rest of our model, but also that are aware of character-level and document-level information, allowing our model to make use of multiple levels of linguistic information. The GRU output encoding for each token has dimensionality 160 (that is, the dimensionality of the hidden GRU states in the forward and backward passes is 80).

4.1.3 Question Summary Vector

Following Hudson and Manning (2018), we use a bidirectional LSTM model to construct the question summary vector. LSTMs behave similarly to the GRU, but maintain two hidden states (c_t, the cell state, and h_t, the hidden state) and use different set of equations at each step, given below (assuming hidden and input dimension of d):

$$f_t = \sigma_g(W^{d\times d}x_t + W^{d\times d}h_{t-1} + b) \tag{4.10}$$

$$i_t = \sigma_g(W^{d\times d}x_t + W^{d\times d}h_{t-1} + b) \tag{4.11}$$

$$o_t = \sigma_g(W^{d\times d}x_t + W^{d\times d}h_{t-1} + b) \tag{4.12}$$

$$\tilde{c_t} = \sigma_g(W^{d\times d}x_t + W^{d\times d}h_{t-1} + b_c) \tag{4.13}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c_t} \tag{4.14}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{4.15}$$

Here, x_t refers to the token embedding input at step t, while c_t and h_t refer to the cell state and hidden state of the LSTM at timestep t, respectively. W, U and b refer to various weight matrices which are learnt jointly with the rest of our model.

After encoding the question using a bidirectional LSTM layer, we concatenate the final hidden states output by both LSTM directions to construct the final summary vector:

$$c^f_{t+1}, h^f_{t+1} = \text{LSTM}_f(x_t, c^f_t, h^f_t) \tag{4.16}$$

$$c^b_{t+1}, h^b_{t+1} = \text{LSTM}_b(x_t, c^b_t, h^b_t) \tag{4.17}$$

$$q = [h^f_T; h^b_0] \tag{4.18}$$

Where LSTM_f and LSTM_b are the forward and backward LSTMs respectively, T is the number of tokens in the question embedding, x_t is the t^{th} question token, and q is the final question vector. Similar to the encoded text, this question vector has dimensionality 160.

4.2 Recurrent Memory, Attention, Composition (MAC) Cell

The core of our model is an adaption of the architecture proposed in Hudson and Manning (2018), which performs multi-step reasoning using a recurrent cell: the memory, attention, composition (MAC) cell. This cell is made up of three core units: the control unit, the read unit, and the write unit. As seen in figure 4.2, each cell takes in and outputs two states: the control state, c_i, which represents the reasoning operation to be performed at step i, and the memory state, m_i, which represents the intermediate answer obtained from the previous i cell steps. We initialise the control state with the question vector q, and the memory state with a learnt parameter vector. Below we describe the calculations performed by each MAC cell in detail. All parameters in each unit except the position-aware parameters in equation 4.19 are shared between all MAC cells. Let d be the hidden state size of the MAC network (which equals 160 in our model).

4.2.1 Control Unit

The control unit takes in the previous control state and outputs a new control state by calculating an attention distribution over all question words. Intuitively, this unit determines what reasoning operations should be performed by the rest of the cell.

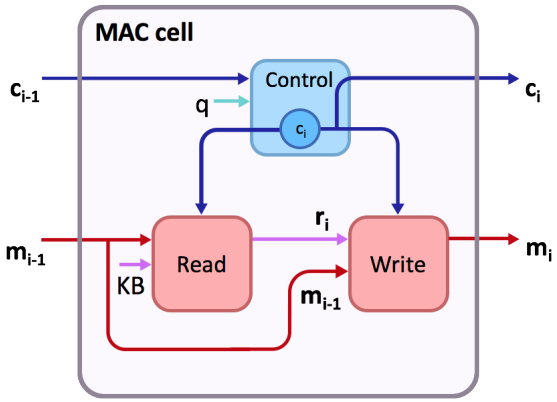


FIGURE 4.2. Diagram of a MAC cell from Hudson and Manning (2018). c_i and m_i represent the control and memory states at step i, while q and KB represent the encoded question and knowledge base. r_i represents the information extracted by the read unit at step i.

First, we convert the question vector into a position-aware form, and combine it with the previous control state:

$$q_i = \tanh(W^{d\times d} q + b) \tag{4.19}$$

$$cq_i = W^{2d\times d}[q_i; c_{i-1}] + b \tag{4.20}$$

We then calculate an attention distribution over each question word using the dot product between cq_i and each encoded token, and use this to construct the next control state:

$$ca_{i,s} = W^{d\times d}(cq_i \cdot cw_s) + b \tag{4.21}$$

$$cv_{i,s} = \text{softmax}(ca_{i,s}) \tag{4.22}$$

$$c_i = \sum_{s=1}^{S} cv_{i,s} \cdot cw_s \tag{4.23}$$

Note that if the encoded tokens cw do not have dimensionality equal to the MAC hidden size d, we simply linearly project them to have dimensionality d. Since $b = d$ for our model, we do not have to do this.

4.2.2 Read Unit

The read unit then utilises the control unit's output to retrieve information from the input document relevant to the current reasoning operation.

First, the previous memory state is used to construct a memory-aware representation of the current document, allowing the cell to discover information that might only be relevant in the context of a prior reasoning step. This is then combined with the original document representation to allow the model to consider information not directly related to the previous step:

$$I_{i,h} = (W^{d\times d} m_i + b) \odot (W^{d\times d} K_h + b) \tag{4.24}$$

$$I'_{i,h} = W^{2d\times d}[I_{i,h}; K_h] + b_I \tag{4.25}$$

In the original MAC cell design, this combined representation I' is then used to compute an attention distribution by calculating a dot product between I' and c_i, the current control state. While this worked well for image-based representations, our experiments found that it performed poorly at identifying important information in the document representations. Instead, we utilise a modified form of bi-attention from Jiang and Bansal (2019a), which combines the attention distribution output by the control unit with the popular bi-attention layer proposed by Seo et al. (2017).

This bi-attention layer operates on a similarity matrix, M, in which each cell M_{sh} represents the similarity score between question token cw_s and document token $I'_{i,h}$:

$$M_{s,h} = W^{d\times d} u_s + W^{d\times d} I'_{i,h} + W^{d\times d}(u_s \odot I'_{i,h}) \tag{4.26}$$

This similarity matrix is then used to calculate the 'context to query' attention, which summarises the most important question words for each document word:

$$p_{s,h} = \frac{exp(M_{s,h})}{\sum_{s=1}^{S} exp(M_{s,h})} \tag{4.27}$$

$$c_{q_h} = \sum_{s=1}^{S} p_{s,h} u_s \tag{4.28}$$

We then compute the 'query to context' attention. While the original bi-attention mechanism used max-pooling over the question words cw_s to identify the most important question word, we instead use the attention distribution calculated by the control unit, $cv_{i,s}$. Thus, the query to context attention now highlights the most important words in the context for the words in the query chosen by the control unit:

$$m'_h = cv_{i,s} \cdot M_{s,h} \tag{4.29}$$

$$p_h = \frac{exp(m'_h)}{\sum_{h=1}^{H} exp(m'_h)} \tag{4.30}$$

$$q_c = \sum_{j=1}^{J} p_h I'_{i,h} \tag{4.31}$$

Finally, we combine the different attentions flows together with I' to construct the final output for the bi-attention layer, and shrink it down to the hidden size of the MAC cell:

$$h'_j = W^{4d \times d}[I'_h; c_{q_h}; I'_h \odot c_{q_h}; c_{q_h} \odot q_c] + b \tag{4.32}$$

Intuitively, h'_j is a question-aware representation of the input document focussed on the parts of the question highlighted by the control unit. This is used as the output for the final MAC cell.

Finally, we compute an attention distribution over h'_j to construct a summary vector of the information retrieved by the read unit:

$$ra_{i,h} = W^{d \times d} h'_j + b \tag{4.33}$$

$$rv_{i,h} = \text{softmax}(ra_{i,h}) \tag{4.34}$$

$$r_i = \sum_{h=1}^{H} rv_{i,h} \cdot K_h \tag{4.35}$$

r_i is then passed to the write unit to construct the memory next state.

4.2.3 Write Unit

The write unit merges the retrieved information r_i and the previous memory state m_{i-1} to form the next memory state m_i. This is simply done by concatenating the retrieved information and control state and passing it through a linear layer:

$$m_i = W^{2d \times d}[m_{i-1}; r_i] + b \tag{4.36}$$

In addition, the control state can also be optionally integrated into the memory state at this point, which is performed in the default MAC setup provided by Hudson and Manning (2018) (although not mentioned in their paper). In this case, equation 4.36 becomes:

$$m_i = W^{3d \times d}[m_{i-1}; r_i; c_i] + b \tag{4.37}$$

Optionally, a gate can be applied to the memory state to allow the previous memory state and current memory state to be merged. This gate utilises the control state to determine how much of the information retrieved at the current cell step should be used for the final memory state:

$$m_i = \sigma(c_i') \cdot m_{i-1} + (1 - \sigma(c_i')) \cdot m_i \tag{4.38}$$

Intuitively, this allows cells to be 'skipped' if a question does not require every cell for its reasoning process.

We utilise both the integration of the control state and the memory gate in our GloVe-based model.

4.3 Output Unit

Finally, we utilise the output from the final MAC cell to make our predictions. First, we take the question summary vector q and final memory output m_i, and concatenate the two to form an answer summary

vector a. We then utilise a to make our prediction of whether the answer to the question is yes, no, or a span:

$$a = W^{2d \times d}[q; m_i] + b_a \tag{4.39}$$

$$O_{\text{t}} = W_{o_1} a + b_{o_1} \tag{4.40}$$

$$\text{pred}_{yns} = \text{softmax}(O_{\text{t}}) \tag{4.41}$$

If the answer is predicted to be 'yes' or 'no', we just output the given word as the answer. If the answer is instead predicted to be a span, we follow the baseline model's output process for predicting the start and end indices of the answer span alongside with supporting facts (which are still predicted in the case of a yes/no answer). This process goes as follows. First, we take the bi-attention output from the final MAC cell and pass it through a self-attention layer to allow for prediction-specific processing and the relocation of attention to the start and the end of the predicted answer span. We also provide a residual connection to the pre-self-attention output. Formally:

$$out = h'_j \tag{4.42}$$

$$out_1 = \text{GRU}(out) \tag{4.43}$$

$$out_2 = \text{self-att}(out_1) \tag{4.44}$$

$$out_3 = W^{d \times d} out_2 + b \tag{4.45}$$

$$out_4 = out + out_3 \tag{4.46}$$

where h'_j is the output from the bi-attention of the final MAC cell from equation 4.32. 'self-att' refers to the self-attention mechanism, which is simply the bi-attention mechanism from Seo et al. (2017), but with out_1 as both inputs (i.e. in place of the question and document).

We then predict the supporting facts by passing this processed output through a bidirectional GRU and constructing sentence embeddings by taking the hidden states from the start and end of the sentence, identical to how we constructed the question summary vector. The summary vectors are then passed

through a linear layer to conver them to logits, and transformed into probabilities using the sigmoid function. Formally, the prediction for sentence i is formed by the following equations:

$$out' = \text{GRU}(out_4) \tag{4.47}$$

$$s_i = [out_i'^f; out_i'^b] \tag{4.48}$$

$$s_i' = W^{d\times 1} s_i + b \tag{4.49}$$

$$\text{pred}_i = \sigma(s_i') \tag{4.50}$$

Where out'^f and out'^b refer to the outputs from the forward and backwards GRUs respectively and σ refers to the sigmoid function. We mark a sentence as a supporting sentence if pred_i is over 0.3.

Finally, we have to predict the span start and end indices. We first incorporate the supporting fact prediction by concatenating the sentence logit s_i' to each word in its respective sentence. That is, if word j is in sentence i, then the output vector representing word j is modified as such:

$$out_j' = [out_j; s_i'] \tag{4.51}$$

We then pass this modified output through a bidirectional GRU, and use the output to produce logits for the start index. The output from the GRU layer is concatenated with the modified output and passed into a second bidirectional GRU to produce logits for the end index. Both logits are passed through the softmax function to produce normalised probabilities of each token being the start and end index:

$$out_s = \text{GRU}(out') \tag{4.52}$$

$$O_s = W^{d \times d} \cdot out_s + b \tag{4.53}$$

$$\text{pred}_s = \text{softmax}(O_s) \tag{4.54}$$

$$out_e = \text{GRU}([out', out_s]) \tag{4.55}$$

$$O_e = W^{d \times d} \cdot out_e + b \tag{4.56}$$

$$\text{pred}_e = \text{softmax}(O_e) \tag{4.57}$$

Finally, we take as answer the span that maximises the probability $\text{pred}_s * \text{pred}_e$, with the constraint that $s \leq e$ (i.e the start token must be before the end token). With this, we have all outputs of our model.

4.4 Loss

We calculate loss values for both answer and supporting fact predictions. First, we use cross-entropy loss for training our yes/no/span, start index, and end index predictions. Formally, if we have c mutually-exclusive classes, predicted probabilities $P = [p_0, p_1, ..., p_c]$, and ground-truth labels $G = [g_0, g_1, ..., g_c]$, where $g_i = 1$ if class i is the ground truth answer, and $g_i = 0$ otherwise, then cross-entropy loss is defined as:

$$\text{CE}(G, P) = -\sum_{i=0}^{i=c} g_i \cdot \log(p_i) \tag{4.58}$$

Note that the probabilities P are commonly computed using the softmax function.

For supporting fact prediction, we cannot utilise this cross-entropy loss as multiple sentences may be supporting facts, and we have no knowledge in advance of the number of supporting fact sentences. In this case, we train each sentence prediction independently using binary cross-entropy loss. Formally, if we have a predicted supporting fact probability s_i, and ground truth value y_i (where $y_i = 1$ if the sentence is a supporting fact, and 0 otherwise), then the binary cross entropy loss is:

$$\mathrm{BCE}(y_i, s_i) = y_i \cdot \log(s_i) + (1 - y_i) \cdot \log(1 - s_i) \tag{4.59}$$

Note that the probabilities s_i are usually computed using the sigmoid function, as we have done above.

The final loss for our model during training is simply given by adding all our loss functions together:

$$L = \mathrm{CE}(G_{yns}, \mathrm{pred}_{yns}) + \mathrm{CE}(G_s, \mathrm{pred}_s) + \mathrm{CE}(G_e, \mathrm{pred}_e) + \mathrm{BCE}(G_{\text{sent}}, S) \tag{4.60}$$

Here G indicates the ground truth for a particular prediction, and S indicates all supporting fact predications. We ignore the start and end index loss (mask it to have value 0) when the ground truth answer is 'yes' or 'no' since in those cases there is no ground truth answer span.

4.5 Optimisation and Training

We jointly optimise all trainable variables in our model using the above loss functions. Following the baseline model, we use stochastic gradient descent (Robbins and Monro, 1951; Kiefer and Wolfowitz, 1952) with a learning rate of 0.1. We use a batch size of 24. We apply early stopping and learning rate reduction during training: we evaluate our model every 1000 steps, and if the predicted answer F1 on the development set is lower than a previous answer F1, we halve the learning rate. When the learning rate reaches 1% of the original learning rate or lower, we stop training and use the model with the best development set answer F1. We find these hyperparameters work best based on tuning experiments performed in chapter 5.

4.6 Conclusion

In this chapter, we have introduced our initial GloVe-based model. This model utilises an augmented MAC cell design to improve its multi-hop reasoning ability. The rest of the model utilises designs from the HotpotQA baseline, thus ensuring that any improvement made by this model is derived from the addition of the MAC cells.

CHAPTER 5

GloVe-based Model Evaluation Results

In this chapter, we evaluate and investigate the performance of our GloVe-based model using the methods outlined in chapter 3. In addition, we explore a naive application of BERT to our model design and investigate the utility of document selection in multi-hop QA.

5.1 Quantitative Evaluation

5.1.1 Performance

We first compare our model's performance against other GloVe-based models. As our training method involves tuning on a dev set, we include both comparisons against the entire dev set of HotpotQA's distractor setting (table 5.1), and then split the dev set in half to construct test and dev splits, and evaluate our model's performance on this constructed test set after tuning on the constructed dev set (table 5.2). For both test and dev sets, our model significantly outperforms the baseline model and is competitive with the Hotpot-NMN model. Despite the Hotpot-NMN model performing better in exact answer matching, the similar answer F1 score indicates our model is just as good at 'fuzzy matching' the answer, but simply less exactly precise. In addition, when removing the auxiliary supervision that the Control + DocQA and Hotpot-NMN models get ('- Bridge Sup'), our model significantly outperforms both models, suggesting our model can learn multihop reasoning without requiring special supervision. Furthermore, as our own model is simply the baseline model with the bi-attention layer swapped out for MAC cells, its improved performance can largely be attributed to the presence of these cells.

We also explore the effect of the number of MAC cells on the performance of our model. While Hudson and Manning (2018) report higher scores on the CLEVR dataset with more cells, the CLEVR dataset differs from the HotpotQA dataset in that it contains questions requiring differing amounts of reasoning steps, where HotpotQA questions always require only two reasoning steps (either the location of the

	Answer		Supporting Fact		Joint	
Model	EM	F1	EM	F1	EM	F1
Baseline	44.44	58.28	21.95	**66.66**	11.56	40.86
Control + DocQA	47.68	-	-	-	-	-
-Bridge Sup	43.31	-	-	-	-	-
Hotpot-NMN	**50.67**	63.35	-	-	-	-
-Bridge Sup	46.56	58.60	-	-	-	-
Hotpot-MAC (ours)	48.82	**63.59**	**23.52**	66.44	**13.54**	**44.57**

TABLE 5.1. Performance of baseline-based MAC model compared with other GloVe-based approaches on HotpotQA distractor dev set. Dashes indicate unreported scores (as in the case of the Control + DocQA model) or that the model was unable to provide the required outputs for those metrics (as in the case of the Hotpot-NMN, which does not output supporting facts).

	Answer		Supporting Fact		Joint	
Model	EM	F1	EM	F1	EM	F1
Baseline	45.46	58.99	22.24	66.62	12.04	41.37
Hotpot-NMN	**49.58**	62.71	-	-	-	-
-Bridge Sup	45.91	57.22	-	-	-	-
Hotpot-MAC (ours)	49.01	**63.81**	**23.41**	**67.90**	**13.10**	**45.21**

TABLE 5.2. Performance of baseline-based MAC model compared with other GloVe-based approaches on the HotpotQA distractor test set[1]. Note that the Control + DocQA model does not evaluate on a test set, and so is left out of this comparison.

bridge entity and then the answer, or the location of the two relevant facts in the case of comparison questions). Thus we hypothesise that only two MAC cells are required for good performance on HotpotQA. This hypothesis is upheld by table 5.3, which shows that 2 cells achieve above-baseline results, and further cells either harm performance or provide minimal benefits. Interestingly, adding these additional cells seems to result in unstable performance, with 3 cells being below baseline performance and 4 and 5 cells being above. Thus we can see that for this task more MAC cells does not necessarily mean improved performance.

Next, we investigate the performance of the Hotpot-MAC and baseline models on the previously-discussed adversarial HotpotQA dataset (Jiang and Bansal, 2019a). We compare the baseline model against the Hotpot-MAC model in table 5.4. Our 2-cell model clearly outperforms the baseline model when trained on the regular HotpotQA training set, but struggles when trained on the adversarial training

[1]For our model, this is our constructed test set mentioned above. For Hotpot-NMN, this is a test set constructed in the same way as ours. For the baseline model, this is the official HotpotQA distractor test set, which is not publicly available. All sets are held out completely during the training of their respective models.

	Answer		Supporting Fact		Joint	
Model	EM	F1	EM	F1	EM	F1
1 cell	37.43	50.65	5.93	37.84	2.69	22.16
2 cell	48.63	63.02	**23.66**	**67.24**	13.23	44.55
3 cell	41.36	55.38	15.88	56.58	8.16	33.76
4 cell	48.22	63.03	20.95	66.17	11.71	43.84
5 cell	**48.82**	**63.59**	23.52	66.44	**13.54**	**44.57**
6 cell	37.15	50.28	8.43	45.90	3.82	25.62
baseline	44.44	58.28	21.95	66.66	11.56	40.86

TABLE 5.3. Performance of Hotpot-MAC and baseline models on HotpotQA distractor dev set. 'X cell' refers to the Hotpot-MAC model using X sequential MAC cells. Highest scores in each column bolded.

Train Set	Reg	Reg	Adv	Adv
Dev Set	Reg	Adv	Reg	Adv
Baseline	58.28	44.51	**62.74**	**70.09**
Hotpot-MAC (1 cell)	50.65	43.32	59.96	66.81
Hotpot-MAC (2 cell)	63.02	58.39	49.95	67.44
Hotpot-MAC (3 cell)	55.38	44.06	60.04	69.17
Hotpot-MAC (4 cell)	63.03	45.59	62.11	68.60
Hotpot-MAC (5 cell)	**63.59**	43.93	57.35	67.33
Hotpot-MAC (6 cell)	50.28	**58.80**	61.53	69.67

TABLE 5.4. Answer F1 of baseline and Hotpot-MAC model across a set of train and development set combinations, using the regular HotpotQA and adversarial HotpotQA distractor sets.

set, with more MAC cells either providing small benefits or worse performance (as before). We hypothesise this is due to the nature of the adversarial dataset: by several fake reasoning paths, a Hotpot-MAC model with only two cells may follow an incorrect path and as a result output the incorrect answer. Thus, more cells are required to allow the model to follow multiple reasoning paths, with the fake reasoning paths acting as noise the model finds difficult to ignore. However, the large gap in performance using the regular training set suggests that the inductive bias of the MAC model does indeed encourage true multi-hop reasoning, even when shortcuts are present in the underlying dataset. In fact, the performance of the baseline model on the regular dev set when trained on the adversarial training set is still below the performance of our Hotpot-MAC model trained on the regular set. Thus it is clear that while the adversarial training set appears to harm the performance of our model, it is still able to learn strong multi-hop reasoning abilities from the regular training set, and is more robust to the adversarial dev set than the baseline model when trained without adversarial data. This strongly suggests the MAC cell design carries a strong inductive bias for multi-hop reasoning.

	SQuAD v1.1		SQuAD 2.0	
Model	EM	F1	EM	F1
DocQA	**72.03**	**80.73**	**61.00**	**63.61**
Hotpot-MAC (2 cells)	71.39	80.47	60.53	63.43

TABLE 5.5. Comparison between DocQA model (Clark and Gardner, 2018) and our model on SQuAD dev sets. DocQA results from our own implementation based on the framework provided by Lee et al. (2019).

Finally, we compare the performance of our multi-hop model with the DocQA model on the SQuAD 1.1 and 2.0 datasets in table 5.5. For the SQuAD 2.0 dataset, which requires an extra no-answer prediction, we utilise the same no-answer module as proposed for the DocQA model (Clark and Gardner, 2018). We find that adding the augmented MAC cells in place of the regular bi-attention layer results in a small drop in performance. These results are similar to those found in Clark and Gardner (2018), who similarly find that their additions to the DocQA model to handle processing multiple paragraphs results in a small performance drop on SQuAD. Therefore, it is clear that the MAC cells contain a strong inductive bias for multi-hop reasoning rather than simply being better at general QA. Furthermore, for general single-hop QA, these cells do not result in a large drop in performance, indicating that the MAC augmented model still retains the vast majority of its regular question answering ability.

5.1.2 Ablations

We next investigate the contribution of each component of our Hotpot-MAC model on the overall performance of the model by ablating each component and measuring the performance of the resulting modified model.

	F1	SP F1	J F1
Hotpot-MAC	63.02	**67.24**	44.55
- control	**64.12**	66.70	**45.27**
- self-attention	54.07	45.87	26.97
- bi-attention	50.22	35.30	20.59
- token output	52.92	37.11	22.19

(a) Ablations on regular data.

	F1	SP F1	J F1
Hotpot-MAC	**67.44**	63.10	**43.56**
- control	65.43	**64.63**	43.51
- self-attention	52.80	46.84	26.57
- bi-attention	64.25	45.15	31.09
- token output	63.94	43.86	30.40

(b) Ablations on adversarial data.

TABLE 5.6. Ablations on the Hotpot-MAC model on the regular and adversarial HotpotQA distractor dev set. See section 5.1.2 for details on each ablation. 'F1', 'SP F1', and 'J F1' refer to answer F1, supporting fact F1, and joint F1 respectively.

We perform ablation experiments using both the regular and adversarial training and development sets, training and testing on the same-paired sets (i.e. reg/reg and adv/adv). The results of our ablation experiments on the regular and adversarial datasets are shown in table 5.6. The ablation setups listed in both tables are as follows:

- **- control**: Removing the control unit from the MAC cell, and replacing the control-based bi-attention with regular bi-attention.
- **- self-attention**: Removing the self-attention layer placed after the MAC cells.
- **- bi-attention**: Removing the control-based bi-attention and using the basic control state - knowledge base interaction in the read unit, as in Hudson and Manning (2018).
- **- token output**: Using the final memory vector m_l as output from the MAC cells, and taking a product between it and the encoded context to highlight the answer location found by the MAC cells before answer prediction.

As we can see, the use of bi-attention is clearly crucial to the modified MAC cell, with its removal causing a significant drop in performance. Furthermore, using just the memory vector as the final output ('- token output'), as done for multiple-choice datasets, also results in a drop in performance. We hypothesise this is because the compressing of the answer text into a single vector is a challenging and difficult task for the model, especially when it then has to re-locate answer candidates for the final output in texts with well over 1000 tokens. We also note the self-attention layer used in the model appears to generally aid performance and particularly aids in the noisy adversarial scenario, where the model must be able to identify and discard multiple possible answer candidates from fake reasoning paths. Especially intriguing is the removal of the control unit, which appears to provide a boost in performance in the regular setup. This suggests that our model is struggling to take advantage of the compositional nature of the questions in HotpotQA, which further examination of the attention maps produced by the control unit also suggest (see section 5.2.2). However, the control unit does seem particularly useful for the adversarial dataset, suggesting the control unit aids with more difficult multi-hop reasoning. As found in Jiang and Bansal (2019b) and Jiang and Bansal (2019a), it is likely that adding auxiliary supervision to train to the control unit would allow the model to make better use of it. Our preliminary experiments found this did not result in greatly improved performance, however.

Thus, while the use of bi-attention in the MAC cell is clearly crucial to its performance, the control unit appears less useful, potentially requiring stronger training to make use of its ability to break down a question. We note that the control state and unit share similarities to query reformulation techniques

used elsewhere in multi-hop QA (Das et al., 2019; Feldman and El-Yaniv, 2019; Shen et al., 2017), which has been previously shown to provide only small improvements in performance (Wang et al., 2019c). As such, while the MAC cell clearly is overall highly useful for multi-hop reasoning, further work is likely required to make good use of the control state.

5.1.3 Parameter Tuning

We investigate if the learning rate or optimizer of the Hotpot-MAC model can be further tuned by optimizing over learning rates and testing the performance of the widely-used Adam optimizer (Kingma and Ba, 2015). As shown in figure 5.1, we find that the original learning rate and optimiser used by the baseline model (SGD optimiser with a learning rate of 0.1) performs best. As such, we used the same learning setup as the baseline model in all the above experiments (with the exception of the SQuAD experiments, for which we copy the setup from the original DocQA paper).

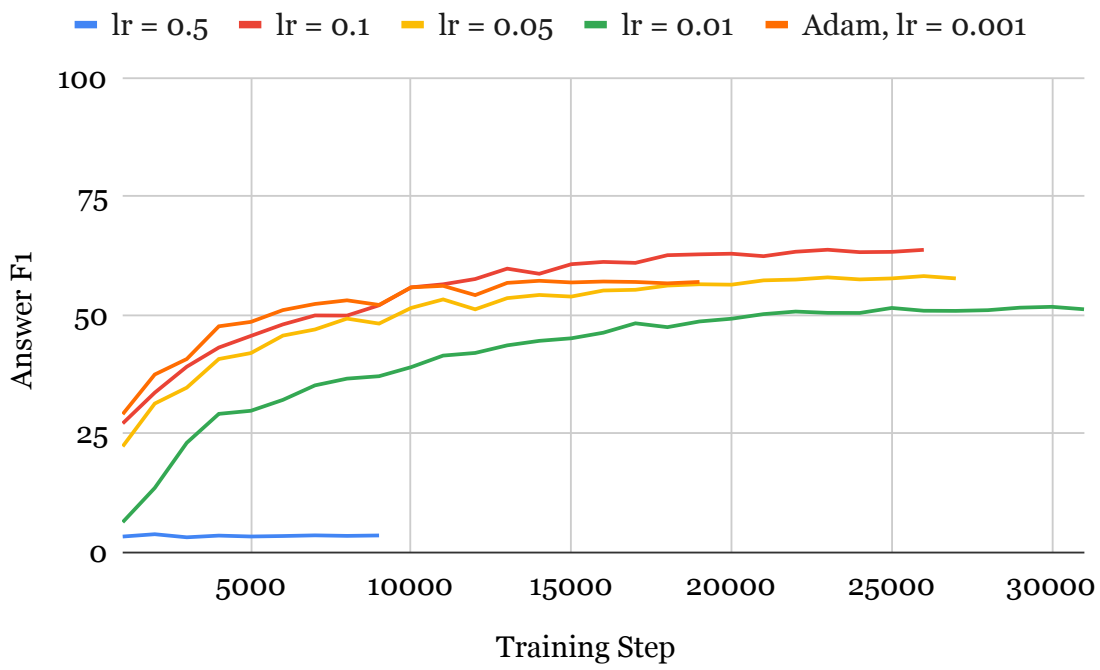


FIGURE 5.1. Answer F1 on HotpotQA distractor dev set against training steps for various hyperparameter setups. 'lr' stands for learning rate. All runs use the SGD optimiser unless otherwise noted in the legend. Lines stop where training ceased based on optimisation scheme outlined in section 4.5.

# Questions	Bridge			Comparison (np)			Comparison (p)		
	5918			1029			458		
	F1	SP F1	J F1	F1	SP F1	J F1	F1	SP F1	J F1
Baseline	57.08	61.45	37.76	53.35	73.42	39.95	**60.48**	80.29	**50.37**
Hotpot-MAC (2-cell)	**64.63**	**63.90**	**44.17**	**56.04**	**77.99**	**44.53**	59.61	**81.62**	50.02

TABLE 5.7. Performance of GloVe-based models broken down by question type on HotpotQA distractor dev set. 'np' and 'p' stand for 'non-polar' and 'polar' respectively. 'F1', 'SP F1', and 'J F1' refer to answer F1, supporting fact F1, and joint F1 respectively.

5.2 Qualitative Evaluation Results

In this section, we investigate the behaviour of our model by examining what types of samples it performs well with and struggles with. We also examine the interpretability of our model by examining its attention maps and compare this with the interpretability of the baseline model. In doing so, we gain a greater understanding of the strengths of our models and identify potential areas for future improvement.

5.2.1 Sample Breakdown

First, we investigate our model's performance across different sample types in HotpotQA. We break down the samples in three ways: by question type, by answer type, and by context length.

First, we examine how well our model does on each question type. As discussed in chapter 3, HotpotQA contains two major question types: *bridge*, which requires finding some bridge entity to find an entity or property of some entity, and *comparison*, which requires comparing two entities. We further distinguish between polar (yes/no) and non-polar comparison questions based on the ground truth answer to the given question. As noted in chapter 3, there are far more bridge questions than comparison questions in HotpotQA, and so we hypothesise that our model will perform better on bridge questions than comparison due to the extra training it receives for bridge questions.

This hypothesis is upheld by the results in table 5.7, which clearly shows the Hotpot-MAC model performs best on bridge-style questions, while still outperforming the baseline on non-polar comparison-type questions. Interestingly, our Hotpot-MAC model outperforms the baseline far more on bridge-style questions than comparison questions. We suggest this is due to the strong inductive bias of the MAC cells lending itself more to bridge questions (where you locate a bridge entity, then use that information to locate the answer) than comparison questions (where you have to retrieve two different facts and

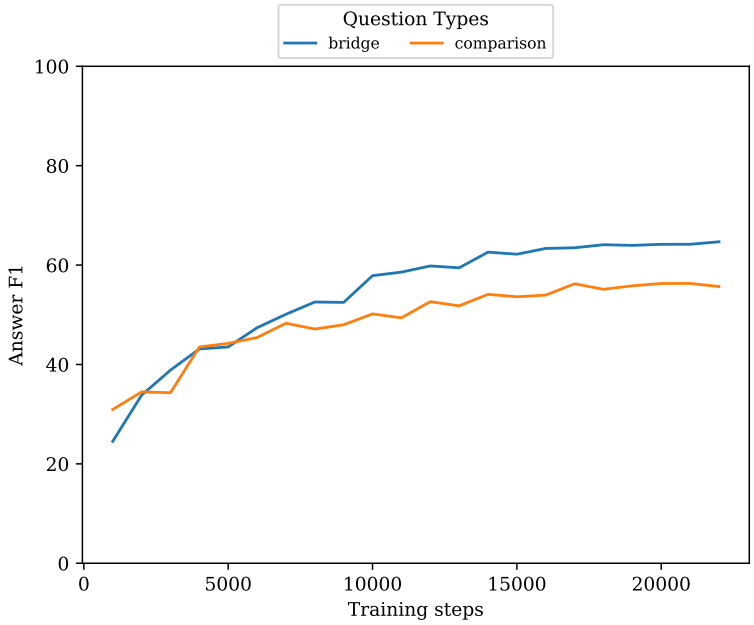


FIGURE 5.2. Training steps against answer F1 for Hotpot-MAC model on HotpotQA distractor dev set, split by question type.

	Baseline	Hotpot-MAC
Actually Correct	10	12
Commonsense	4	2
Discrete Reasoning	10	12
Mislabel	2	3
General	36	46
Multi-hop	36	22
No Answer	2	2
Superspan	0	1

TABLE 5.8. Number of errors made by baseline and Hotpot-MAC model from a sample of 100 errors from HotpotQA distractor dev set. See Appendix A for details on the error types.

combine them in memory). As we saw in the previous section, this inductive bias is strong for multi-hop questions, and this shows it seems to be strongest for bridge-style multi-hop questions. Furthermore, the relative scarcity of comparison questions in the dataset may also mean the model struggles to learn how to deal with them as well as bridge questions - as we can see in figure 5.2, it is only after some time that our model starts predicting bridge questions better than comparison questions. An additional comparison of the number of different types of errors made by the baseline and Hotpot-MAC model from 100 random samples in table 5.8 shows that our model makes fewer errors in multi-hop reasoning, providing further evidence that the MAC cell has a strong inductive bias for multi-hop reasoning.

	# Questions	Baseline	Hotpot-MAC
Number	23	**71.26**	67.00
Date	27	**68.20**	66.34
Group	27	46.19	**65.19**
Artwork	16	66.07	**75.00**
Person	73	59.35	**68.86**
Event	6	33.15	**52.97**
Location	54	**58.04**	57.24
Adjective	9	**68.15**	51.83
Proper noun	27	**62.25**	56.40
Common noun	15	21.26	**25.49**
Yes/No	21	61.91	**66.67**
Mislabel	2	0.00	**6.67**

TABLE 5.9. Answer F1 for different answer types for our GloVe-based models on HotpotQA distractor dev set.

Next, we examine the performance of our model based on answer type, using the annotated answers described in chapter 3. As seen in table 5.9, our model largely outperforms the baseline at answers that are entity names (artworks, company names, people's names, event names, etc.), while under-performing compared to the baseline at other answer types, such as those involving numbers and adjectives. We again hypothesise this is due to the design of the MAC network: the sequential cell design is likely best at linking entities from one part of the text to another part, rather than the discrete or numerical reasoning required by some questions.

Finally, we examine the effect of context length on the performance of our model. Dealing with long text is a challenge for many NLP models, including the state-of-the-art. Furthermore, unlike current BERT-based state of the art models, which often use a document selection step to shrink the input to their core question answering models, both the baseline and our baseline-based model take in all ten documents in the distractor setup concatenated together. As such, it is well worth examining the effect of these larger text sizes on our model.

As seen in figure 5.3, both the baseline and our model exhibit similar behaviour with context length: they are largely robust to increasingly long contexts until around 12000 characters, at which point they completely collapse. This is likely due to the relative scarcity of such long contexts in the dataset: as seen in figure 5.4, contexts longer than 10000 characters are uncommon in the HotpotQA training data. Thus, it is clear that neither the baseline nor Hotpot-MAC model struggle with the extremely long contexts present within the HotpotQA setup.

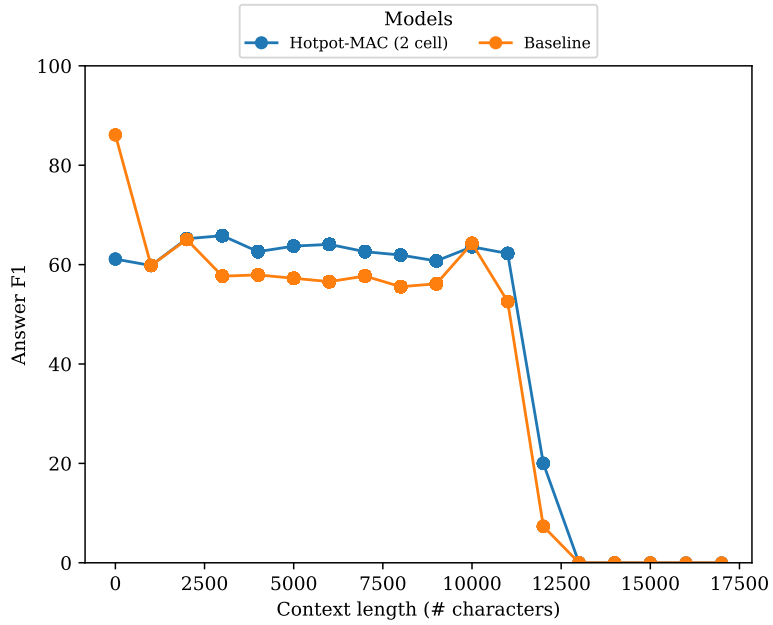


FIGURE 5.3. Context length against answer F1 for baseline and Hotpot-MAC models on HotpotQA distractor dev set. Datapoints constructed by rounding all context lengths to nearest 1000 and averaging F1 of points with same rounded length. Note that bins are not necessarily of the same size.

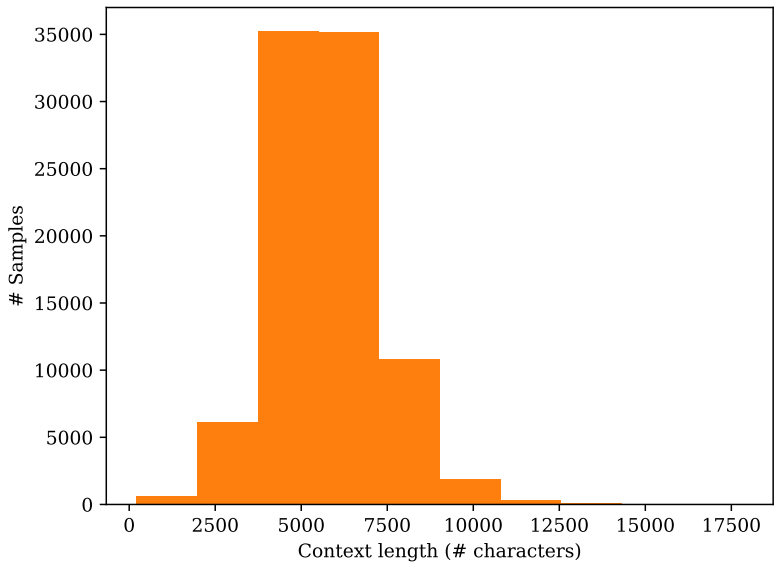


FIGURE 5.4. Histogram of context lengths in the HotpotQA distractor setting training set.

Thus, we have shown that our novel model design specifically carries a strong inductive bias for bridge-type multi-hop questions, while still demonstrating strong performance on comparison questions. Furthermore, it makes fewer multi-hop errors than the baseline model, and is robust to context length, with little degradation in performance when processing long contexts.

5.2.2 Attention Maps

One of the potential benefits of the MAC architecture is increased interpretability through the attention maps generated by each cell. Hudson and Manning (2018) show when introducing the MAC network that interpretable attention maps naturally arise when training on the CLEVR dataset. However, we find that such interpretable maps do not appear in our Hotpot-MAC model, although examining these maps still provides some insight on the interior workings of the model. We visualise the control attention (attention distribution over the question words calculated in the control unit) and read attention (attention distribution over the context words calculated in the read unit).

We visualise the attention maps of several questions from the HotpotQA dev set, with the full attention maps (along with question ids) given in appendix B. Darker colors indicate higher attention values and thus more 'importance' given to that word in a given MAC cell unit. We first note that while the control attention maps do change across cells, they do not always follow an intuitive path. For example, comparison questions do not seem to iteratively focus on the entities in the questions:

Control Attention Question ID 5a8b57f25542995d1e6f1371

Cell #1: Were Scott Derrickson and Ed Wood of the same nationality ?

Cell #2: Were Scott Derrickson and Ed Wood of the same nationality ?

While we would expect the model to iteratively focus on the two names ('Scott Derrickson' and 'Ed Wood'), it never focuses on 'Ed Wood'. Examining the read unit attention maps further shows the model focusing on nationalities (mainly 'American') in both cells, providing little indication of a reasoning process.

Similarly, when examining bridge questions, we do not find entirely intuitive attention maps, although the reasoning performed by the model is made somewhat clearer:

Control Attention Question ID 5abd94525542992ac4f382d2

Cell #1: 2014 S / S is the debut album of a South Korean boy group that was formed by who ?

Cell #2: 2014 S / S is the debut album of a South Korean boy group that was formed by who ?

While the reasoning path here is opposite to what we would expect , as it first focuses on ‘formed by who’, and then the album name, it is still clearer than in the comparison question example above. Examining the attention maps of the read unit for this sample further displays the reasoning path taken by our model: first, the various companies that form South Korean boy groups are highlighted, and then after considering evidence from the rest of the question, the answer (‘YG entertainment’) is further highlighted with more confidence than other potential answers[2]:

Read Attention Question ID 5abd94525542992ac4f382d2

Cell #1: <t> List of awards and nominations received by Shinee </t>... The group was formed by S.M. Entertainment in 2008 ... <t> Cho Kyuhyun </t> Cho Kyu - hyun (born February 3 , 1988) , better known mononymously as Kyuhyun ... a former member of the South Korean ballad group S.M. the Ballad . ... 2014 S / S is the debut album of South Korean group WINNER . It was released on August 12 , 2014 by the group ’s record label , YG Entertainment . ... a South Korean boy group formed by LOEN Entertainment in 2013 . ... Winner (Hangul : 위너) , often stylized as WINNER , is a South Korean boy group formed in 2013 by YG Entertainment and debuted in 2014 . ... Madtown (Hangul : 매드타운) , often stylized as MADTOWN , is a South Korean boy group formed in 2014 by J. Tune Camp . The group consists of Moos , Daewon , Lee Geon , Jota , Heo Jun , Buffy...

Cell #2: <t> List of awards and nominations received by Shinee </t> ... The group was formed by S.M. Entertainment in 2008 ... <t> Cho Kyuhyun </t> Cho Kyu - hyun (born February 3 , 1988) , better known mononymously as Kyuhyun ... a former member of the South Korean ballad group S.M. the Ballad . ... 2014 S / S is the debut album of South Korean group WINNER . It was released on August 12 , 2014 by the group ’s record label , YG Entertainment . ... a South Korean boy group formed by LOEN Entertainment in 2013 . ... Winner (Hangul : 위너) , often stylized as WINNER , is a South Korean boy group formed in 2013 by YG Entertainment and debuted in 2014 . ... Madtown (Hangul : 매드타운) , often stylized as MADTOWN , is a South Korean boy group formed in 2014 by J. Tune Camp . The group consists of Moos , Daewon , Lee Geon , Jota , Heo Jun , Buffy...

[2]Note above we have removed sections of the text with low attention values to improve legibility. The full attention map can be found in appendix B.

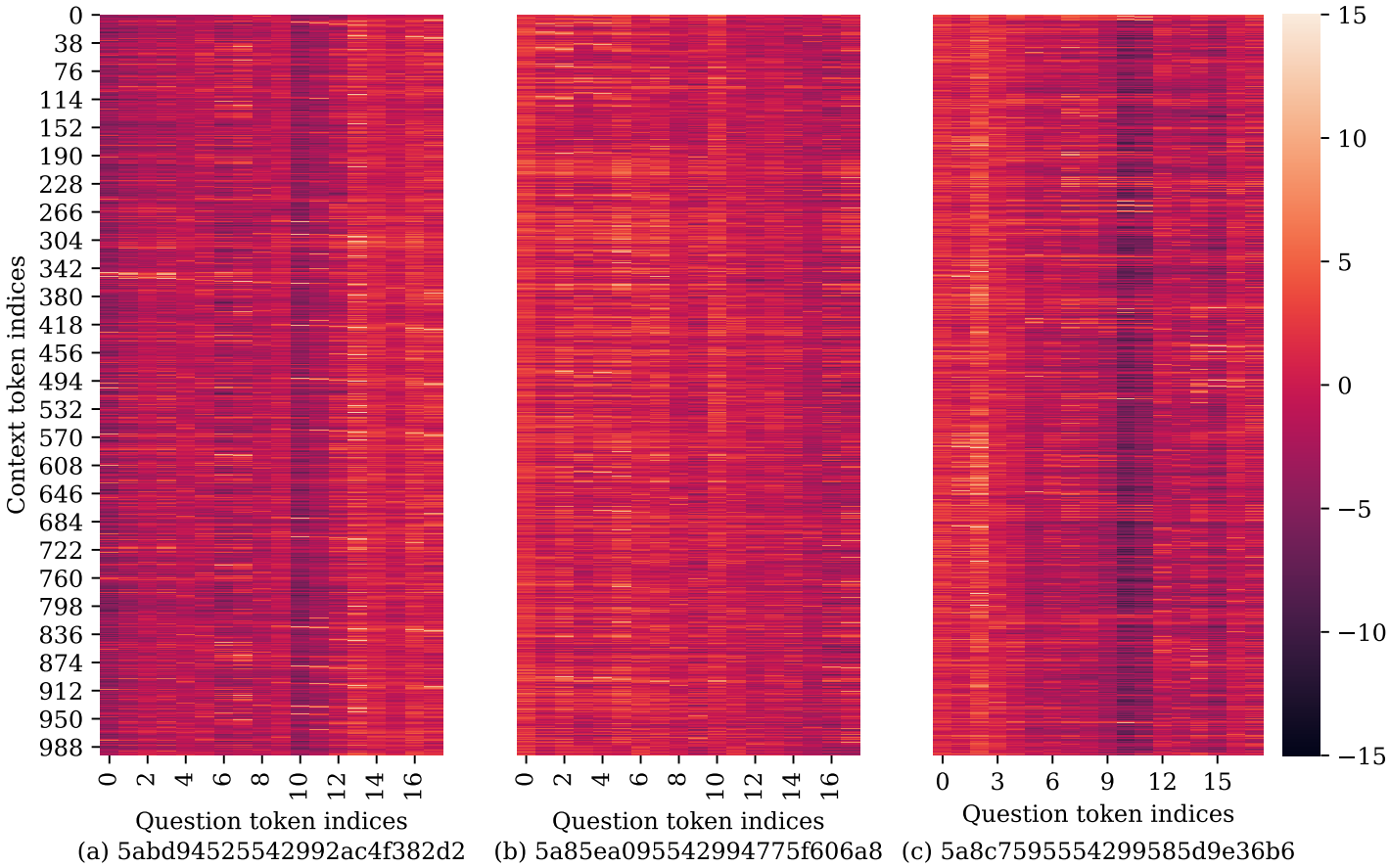


FIGURE 5.5. Heatmap of unnormalised logits from attention matrix of question-context bi-attention layer in baseline model for several questions from HotpotQA. Token indices rather than text given for legibility, and logit values clipped to the range [-15, 15]. Question IDs given below each heatmap.

While these attention maps are not entirely interpretable, we now compare them to the attention maps produced by the baseline. As the baseline makes use only of bi-attention, a standard way of visualising this is via displaying the attention matrix that underlies both the query to context and context to query attention mechanisms. However, the long texts in HotpotQA make this attention matrix hard to understand. For example, we show the full bi-attention matrix for three samples in figure 5.5 - it is difficult to parse at a glance here what words are important, let alone even space out each context word such that they are readable. This stands in contrast to the MAC cell's more readily interpretable attention, where only one value is associated with every word, as we have seen above.

However, we better can visualise part of the baseline's attention matrix by 'flattening' the attention matrix and showing which document words are considered most salient in the query to context attention, which shows which document words are most relevant to the question (see chapter 4 for more details):

Question ID 5abd94525542992ac4f382d2

Question-Context Attention: <t> 2014 S / S </t> 2014 S / S is the debut album of South Korean group WINNER . It was released on August 12 , 2014 by the group 's record label , YG Entertainment . The members were credited for writing the lyrics and composing the majority of the album 's songs . <t> History (band) </t> History (Korean : 히스토리) was a South Korean boy group formed by LOEN Entertainment in 2013 . They debuted on April 26 , 2013 with " Dreamer " , featuring the narration of their labelmate IU . They were LOEN Entertainment 's first boy group . They officially disbanded on May 12 , 2017 . <t> Winner (band) </t> Winner (Hangul : 위너) , often stylized as WINNER , is a South Korean boy group formed in 2013 by YG Entertainment and debuted in 2014 . It currently consists of four members , Jinwoo , Seunghoon , Mino and Seungyoon . Originally a five - piece group with Taehyun , who later departed from the group in November 2016 . <t>

Context-Context Self-Attention: <t> 2014 S / S </t> 2014 S / S is the debut album of South Korean group WINNER . It was released on August 12 , 2014 by the group 's record label , YG Entertainment . The members were credited for writing the lyrics and composing the majority of the album 's songs . <t> History (band) </t> History (Korean : 히스토리) was a South Korean boy group formed by LOEN Entertainment in 2013 . They debuted on April 26 , 2013 with " Dreamer " , featuring the narration of their labelmate IU . They were LOEN Entertainment 's first boy group . They officially disbanded on May 12 , 2017 . <t> Winner (band) </t> Winner (Hangul : 위너) , often stylized as WINNER , is a South Korean boy group formed in 2013 by YG Entertainment and debuted in 2014 . It currently consists of four members , Jinwoo , Seunghoon , Mino and Seungyoon . Originally a five - piece group with Taehyun , who later departed from the group in November 2016 .

Note above we truncated the above text to just supporting documents (and one extra). As we can see, the attention from the baseline model is much noisier, and as such, it is much harder to determine the reasoning paths taken by the model. It is also important to note that this visualisation only displays half

the bi-attention mechanism. Visualising the context to query attention, where an attention distribution is calculated over the question word for every context word, would require a 2D heatmap similar to figure 5.5.

Thus, while the MAC cell is certainly not as interpretable with this task when compared to VQA, it certainly improves over the baseline model, providing much sparser and thereby more interpretable attention maps, even if the reasoning processes followed are not entirely intuitive. Training the model to follow specific reasoning paths, as done in Jiang and Bansal (2019a), may aid in further interpretability.

5.3 Utilising BERT

5.3.1 A Naive Approach

While our model performs well with GloVe-based embeddings, its performance lags greatly behind the current state-of-the-art, which relies on BERT (and BERT-like) pretrained models for high performance (Asai et al., 2020; Fang et al., 2020; Tu et al., 2020; Qiu et al., 2019; Dhingra et al., 2020). We first investigate BERT integration by simply replacing the GloVe and character-based embeddings with BERT, keeping the rest of the model the same (i.e. replacing the purple modules in figure 4.1 with BERT). As BERT cannot process all 10 documents in the distractor setting at once, we utilise a pretrained document selection model from the SAE model (Tu et al., 2020), which ranks and selects the top two documents at test time. We concatenate the two chosen documents along with the question and pass them through BERT to generate contextually-aware embeddings for both the question and documents.

We also design a naive BERT baseline model by utilising the supporting fact prediction design from the HotpotQA baseline for supporting fact prediction. This utilises the hidden outputs from the final BERT layer to perform supporting fact prediction and predicts the answer location directly from BERT's hidden outputs (whilst also using the SAE document selection model to reduce the input to BERT). For both the MAC and baseline BERT-based models, we utilise the Adam optimiser with a learning rate of 10^{-5}, using the same learning rate reduction strategy as the GloVe model. We compare these two models with both GloVe-based and other BERT-based models in table 5.10.

As we can see, while using BERT provides a large boost across all metrics compared to the GloVe-based models, it appears that MAC network adds little to no boost over the already powerful reasoning ability of BERT. Further augmentations to the MAC network, including extra supervision on locating

	Answer		Supporting Facts		Joint	
Model	EM	F1	EM	F1	EM	F1
HotpotQA Baseline*	44.44	58.28	21.95	66.66	11.56	40.86
Hotpot-MAC (2 cell)*	48.63	63.02	23.66	67.24	13.23	44.55
BERT Baseline	60.08	74.26	60.32	86.28	40.30	66.61
Hotpot-MAC + BERT (2 cell)	60.49	74.66	59.92	86.35	40.66	66.85
+ bridge sup.	60.58	**75.01**	59.97	86.32	40.51	**67.12**
+ 8 cells	60.41	74.64	**60.54**	**86.61**	**40.96**	67.06
SAE (Tu et al., 2020)	**61.32**	74.81	58.06	85.27	39.89	66.45
HGN (Fang et al., 2020)	-	74.26	-	**86.61**	-	66.90

TABLE 5.10. Performance of MAC model with BERT alongside our BERT baseline and state-of-the-art models for HotpotQA distractor dev set. Models marked with * use GloVe for text encoding, while all other models use BERT-base-uncased. Dash indicates scores not available.

the bridge entity[3] ('+ bridge sup' in table 5.10) or using 8 cells instead of 2 ('+ 8 cells' in table 5.10), seem to provide little to no benefit over the baseline. Examining the attention cells of these MAC cells shows that the cells only focus on the predicted answer (or bridge entity, if trained to detect the bridge entity), further suggesting that they are simply 'passing on' the predictions of the BERT model rather than performing reasoning themselves. However, with the bridge supervision, we are able to an achieve answer F1 score slightly above the current state of the art in HotpotQA, suggesting that this extra supervision does bring some benefit, albeit small, potentially due to it more directly encouraging multi-hop reasoning within the model, thereby making it easier for the model to find the correct answer.

We also note in table 5.10 that the existing state-of-the-art on HotpotQA does little to improve on our naive BERT baseline, suggesting that the extra methods used by these approaches ontop of BERT add little to the performance of the overall models. This is especially true for the SAE model, which utilises a complex graph-based model for predicting supporting facts that under-performs our simple GRU-based supporting fact predictor. This matches the findings of Shao et al. (2020), who find that the graph methods utilised in the DFGN model are largely unnecessary when correctly fine-tuning BERT. We note that this behaviour may not hold for larger pretrained models such as RoBERTa-large, which models such as the HGN appear to provide some (small) performance boosts over. Ultimately, however, it is clear that the most important parts of all these models are the document selection step (since incorrect document

[3]This extra supervision is performed by training the first of the two MAC cells to output the start location of the bridge entity, which is detected via the same method as Jiang and Bansal (2019b) - if the title of the answer document is in the title of a supporting document, that is marked as the bridge entity. If the answer appears in both supporting documents, we check if the title of one exists in the other and use that title. If we cannot detect a bridge entity, we do not train the bridge detection on that sample.

	Answer		Supporting Facts		Joint	
	EM	F1	EM	F1	EM	F1
BERT baseline	**61.89**	**76.34**	**64.42**	89.77	**42.78**	**69.92**
- separately encoded	61.55 (-0.37)	76.00 (-0.34)	61.97 (-2.45)	88.85 (-0.92)	41.00 (-1.78)	68.76 (-1.16)
Hotpot-MAC + BERT	61.63	76.18	64.39	**89.95**	42.70	69.74
- separately encoded	61.30 (-0.33)	75.66 (-0.52)	62.73 (-1.66)	89.23 (-0.72)	41.23 (-1.47)	68.58 (-1.16)

TABLE 5.11. Performance of BERT baseline and Hotpot-MAC with BERT on HotpotQA distractor dev set when utilising jointly encoded documents and separately encoded documents ('-separately encoded'). Input documents are gold documents as annotated in the HotpotQA dataset. Hotpot-MAC model uses 2 cells.

	Answer		Supporting Facts		Joint	
	EM	F1	EM	F1	EM	F1
BERT baseline	60.08	74.26	**60.32**	86.28	40.30	66.61
- separately encoded	59.62 (-0.46)	73.90 (-0.36)	58.07 (-2.25)	84.54 (-1.74)	38.68 (-1.62)	65.52 (-1.09)
Hotpot-MAC + BERT	**60.49**	**74.66**	59.92	**86.35**	**40.66**	**66.85**
- separately encoded	59.39 (-1.10)	73.56 (-0.36)	58.58 (-1.34)	85.77 (-0.58)	38.65 (-2.01)	65.33 (-1.52)

TABLE 5.12. Performance of BERT baseline and Hotpot-MAC with BERT on HotpotQA distractor dev set when utilising jointly encoded documents and separately encoded documents ('-separately encoded'). Input documents are documents selected by the SAE mechanism. Hotpot-MAC model uses 2 cells.

selection naturally harms downstream performance) and the underlying pretrained BERT model, with other aspects of these models adding at best minimal performance improvements.

5.3.2 Can MAC and BERT Work Together?

Given that the MAC cell is unable to add much performance over BERT when applied in a naive way, we explore one potential method to force the MAC cell to perform multi-hop reasoning with BERT: we process each document separately with BERT[4], and then concatenate the outputs and pass them through our augmented MAC cells to predict the answer location. We compare this with a basic model where the documents are processed separately with BERT and then concatenated and passed directly to answer prediction layers. We provide the performance of these two models on the gold document set (i.e. only the documents annotated in the HotpotQA dataset as required for answering a given question) and the SAE selected document set in tables 5.11 and 5.12 respectively.

[4]We still process each document concatenated with the question with BERT in order to provide rich question-aware representations of the input documents, as we found separate processing of the question and document performed relatively poorly.

	Answer		Supporting Facts		Joint	
	EM	F1	EM	F1	EM	F1
RoBERTa-large	**69.80**	**83.39**	**67.54**	**90.99**	**49.53**	**76.75**
- separately encoded	67.28 (-2.52)	81.34 (-2.05)	65.28 (-2.26)	90.23 (-0.76)	45.86 (-3.67)	74.24 (-2.51)
Hotpot-MAC + R-L	69.42	83.12	67.45	91.07	49.07	76.58
- separately encoded	67.01 (-2.41)	80.94 (-2.18)	65.16 (-2.29)	90.09 (-0.98)	45.82 (-3.25)	73.85 (-2.73)

TABLE 5.13. Performance of RoBERTa-large-based models on HotpotQA dev distractor set when using joint and separate document encoding. Only gold documents are used in evaluation and training. 'R-L' is short for 'RoBERTa-large'.

Notably, removing the joint encoding results in lower performance across the board, with especially large drops in supporting fact performance, suggesting that cross-document encoding aids in locating relevant facts for the answer. This is likely due to the fact that supporting sentences include those that link entities across documents, which is naturally difficult to determine when the two documents are separately encoded. Utilising MAC cells for reasoning across the documents seems to help slightly, reducing the drop that comes from separately encoding documents, but still does not perform as well as the BERT baseline with jointly encoded documents. However, the answer-only metrics, especially F1, only suffer smaller drops in performance when not jointly encoding documents, indicating that locating the answer to a given question does not require cross-document reasoning for strong performance. These results are in line with Wang et al. (2019c), who show a similarly small boost from adding cross-document reasoning on top of jointly encoded documents. Furthermore, these results support the claims of Min et al. (2019) and Jiang and Bansal (2019a), who show that the HotpotQA dataset contains lexical shortcuts that can be exploited by models to avoid multi-document reasoning. Both works suggest fixing the dataset by adding stronger distractor documents, requiring the model to be better at selecting the relevant documents for answering the question. However, our results show that given strong document selection, there is little benefit from cross-document reasoning, suggesting that existing complex methods for multi-hop QA, which often involve first selecting two documents and then constructing complex graph networks, rely far more heavily on that initial (and often under-examined) document selection step than the rest of the design.

However, as seen in table 5.13, we find that the gap in performance between separate and joint encoding is much larger when using the RoBERTa-large model (Liu et al., 2019), with further analysis showing this gap largely arises from joint encoding performing far better (+ 6 points in answer F1) at non-polar comparison questions. This makes sense: only through joint encoding would the model be able to compare and select the two entities in question. Such comparison questions, though, can actually

be ‘solved’ through document selection as well: if a model is able to select the document containing the answer entity alone, it will naturally then provide the correct answer. Adding MAC cells does not change this behaviour, nor does it provide a boost in performance. Ultimately, this analysis shows that existing models only utilising BERT rely more heavily on document selection methods than their complex graph-based designs, and only larger BERT-like models are actually able to gain substantial improvements utilising joint encoding. Thus, we now turn our focus to utilising the MAC cell design for document selection, since we have shown it provides little benefit in the naive designs explored above.

5.4 Conclusion

In this chapter, we have thoroughly examined the performance and behaviour of our GloVe-based model. We have shown that it performs significantly above the existing HotpotQA baseline and is competitive with other modular approaches that utilise extra supervision in training, and that our design can be applied to single-hop QA datasets with minimal decrease in performance, despite not being design for this task. In addition, we have explored the behaviour of our model, showing that it provides a strong inductive bias for bridge-type questions and contains more readily interpretable attention maps than the existing baseline model. This all provides strong evidence for the utility of MAC cells in multi-hop QA. Finally, we examined a naive application of BERT to our model design and showed that while BERT-based models do not benefit from MAC cells, they also do not effectively utilise cross-document reasoning, instead relying heavily on an initial document selection step. In the next two chapters, we thus explore a method for utilising MAC cells for document selection, making use of the strengths of the MAC cell and BERT models showcased in this chapter.

CHAPTER 6

BERT-based Model

In this chapter, we introduce our BERT-based model, which integrates the document selection and answer location steps of a multi-hop QA model by utilising the memory of a MAC cell to iteratively select documents, ending with the answer document and prediction. This model design focuses on using the strengths of the MAC cell for the task of document selection, which we have shown is core to competitive performance in multi-hop reasoning. Unless otherwise stated, we utilise the 'bert-base-uncased' model available within Huggingface's transformers library (Wolf et al., 2019) when referring to a 'BERT model'. However, our design is agnostic to choice of underlying pretrained language model, so long as hidden representations can be output for each token along with a document summary vector. This allows us to easily swap in more performant models such as ALBERT (Lan et al., 2020) or RoBERTa (Liu et al., 2019) in place of BERT without modifying the rest of our architecture.

This model retains the same three steps as the GloVe-based model, but the nature of each step is different:

- **Encoding unit**: A BERT model is used to encode the documents (concatenated with the question). The model is also used to generate dense summary vectors of each document for scoring. Finally, we generate a question summary vector using a self-attention mechanism.
- **Recurrent Memory, Attention, Composition (MAC) Cell**: We expand the MAC cell to first rank and score documents before then choosing the highest-ranked one to read. This is performed iteratively, with the final cell output being passed to the output unit. We experiment with using beam search to expand the search space of the cells. In addition, each cell also now directly predicts supporting facts.
- **Output unit**: Finally, we use the outputs from the final MAC cell to predict an answer and output a reranking score, used with beam search to allow the model to explore multiple answers and choose the most likely answer out of those explored.

We provide a high-level diagram of our BERT-based model in figure 6.1.

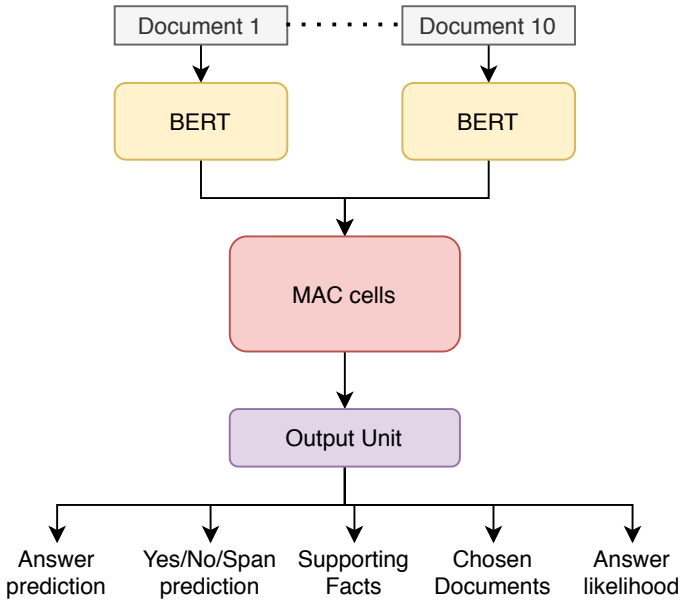


FIGURE 6.1. High-level architecture diagram of our BERT-based model.

6.1 Encoding Unit

6.1.1 Text Preparation

Before feeding text into BERT, we must construct a string utilising a special format that BERT was trained to expect. For question answering, this format is: '[CLS] Question [SEP] Document [SEP]', where [CLS] and [SEP] are special tokens used for classifying and separating sequences respectively. Rather than concatenate all documents together, we construct a string of this format for each document and pass each one through the BERT model separately. This is due to the fact that BERT's input size limitations and computational complexity means encoding extremely long texts (such as all the provided documents concatenated together) is infeasible.

6.1.2 Wordpiece Tokenisation

We next split our text into individual tokens, which we will map to vectors. Rather than simply splitting on whitespace, we utilise the wordpiece algorithm (Wu et al., 2016), which splits the text into sub-word units. For example, the first few lines of book 7 of the Odyssey would be split (or 'tokenised') as such:

'So noble long-suffering Odysseus lay there, conquered by weariness and sleep'

`so, noble, long, -, suffering, o, ##dy, ##sse, ##us, lay, there, ', ', conquered, by, wear, ##iness, and, sleep`

Note that as part of the tokenisation process, we also lower-case all text. Tokens that are not the start of a new word are prefixed with '##'. Rather than use a custom list of tokens to split every word into, we use the list of 30,000 tokens constructed in Devlin et al. (2019), which has proven effective for general English-based NLP tasks. By splitting every word into sub-word units, our model can handle previously unseen words by simply splitting them into text chunks it has seen before, thus still making use of information learnt during training. If this is not possible, the word (or sub-word unit) is instead mapped to a special token, 'UNK', which represents an unknown token. Use of the UNK token is rare but does occur when in some cases, e.g. when dealing with non-English text. In addition, we also mark which tokens belong to the question and which belong to the document using a binary vector, wherein '0' indicates the token belongs to the question and '1' indicates it belongs to the document. This vector, called a segment embedding, allows BERT to better distinguish between the two sequences of text (Devlin et al., 2019).

As we wish to perform batch processing with our model (i.e. train it on multiple queries at once), we have to pad out the tokenized text such that all text within one batch has the same length. We do this by simply appending a special token '[PAD]', onto the end of each tokenized sequence until they are all the length of the longest sequence in the batch. We mark the location of these padding tokens for use throughout our model to ensure that padding tokens are not given any weight, as they contain no useful information for our task at hand.

Finally, we add positional encodings to our model. As BERT is a purely attention-based model, it has no native notion of sequence order (unlike a recurrent neural network). We thus inject positional information via learnt embeddings. The positional encodings are simply summed with the word embeddings (the vectorised tokens) and given by a mixture of sine and cosine functions:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{6.1}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{6.2}$$

Where pos is the token position and i is the dimension we are providing a value for.

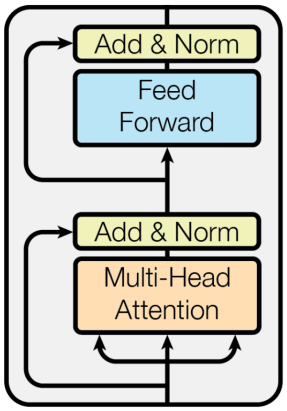


FIGURE 6.2. An encoder block from the standard transformer architecture, from Vaswani et al. (2017).

6.1.3 Transformer-based Encoding

Finally, we encode our embedded text using a series of pre-trained transformer encoder blocks. A transformer encoder block (Vaswani et al., 2017) consists of two key components: a multi-head attention mechanism, and a simple feed-forward layer. In addition, both components utilise layer normalisation (Ba et al., 2016) and residual connections (He et al., 2016) to aid training and stability. An overview of a transformer encoder block is given in figure 6.2.

The first component of the block, multi-head attention, is simply several self-attention mechanisms applied in parallel in order to allow the model to jointly attend to multiple aspects of the input text at once. Formally, if we have a sequence $X = [x_0, x_1, ..., x_n]$ then multi-head self-attention is as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{6.3}$$

$$Q_i = W_i^Q \cdot X \tag{6.4}$$

$$K_i = W_i^K \cdot X \tag{6.5}$$

$$V_i = W_i^V \cdot X \tag{6.6}$$

$$\text{head}_i = \text{Attention}(Q_i, K_i, V_i) \tag{6.7}$$

$$\text{MultiHead}(X) = [\text{head}_0, \text{head}_1, ..., \text{head}_h] \cdot W_h \tag{6.8}$$

Where h is the number of 'heads' in the multi-head attention mechanism and d_k is the dimensionality (size) of each item in X (this scaling effect has empirically been found to aid in avoiding exploding gradients). The second component of the block, a feed-forward layer, is simply two linear layers with a ReLU (Nair and Hinton, 2010) activation between them:

$$\text{ReLU}(x) = \max(0, x) \tag{6.9}$$

$$\text{FFN}(x) = \text{ReLU}(x \cdot W_1 + b_1) \cdot W_2 + b_2 \tag{6.10}$$

The layer normalisation step applied after each component is simply the normalisation of each item in X using learnt scaling factors γ and β. Formally, if we have $x_i \in X$, where x_i is a D-dimensional vector, then it is normalised as such:

$$\mu_i = \frac{1}{D}\sum_{d}^{D} = 0x_i \tag{6.11}$$

$$\sigma_i^2 = \frac{1}{D}\sum_{d}^{D} = 0(x_i - \mu_i)^2 \tag{6.12}$$

$$x_i' = \frac{x_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} \tag{6.13}$$

$$x_i'' = \gamma x_i' + \beta \tag{6.14}$$

$$= \text{LayerNorm}(x_i) \tag{6.15}$$

Where x_i'' is our output value and ϵ is a small value to avoid division by 0. Layer normalisation is used to avoid exploding or diminishing hidden values in recurrent neural networks, as it re-normalises values at each step in the network.

Thus, putting it all together, a transfomer encoder block with input sequence X is formally given by:

$$A = \text{MultiHead}(X) \tag{6.16}$$

$$A' = \text{LayerNorm}(X + A) \tag{6.17}$$

$$L = \text{FFN}(A') \tag{6.18}$$

$$\text{out} = \text{LayerNorm}(L + A') \tag{6.19}$$

There are two main BERT variants (Devlin et al., 2019), utilising different numbers of stacked encoder blocks, heads, and hidden dimensions:

- **BERT-base** utilises 12 blocks, 12 heads, and has a hidden dimension of 768.
- **BERT-large** utilises 24 blocks, 16 heads, and has a hidden dimension of 1024.

In addition to these two models, there exist other BERT variants, usually differentiated by their training mechanisms. We also utilise the **RoBERTa** (Liu et al., 2019) model, which shares the same architecture as BERT but utilises a better-designed pre-training scheme. Like BERT, it comes in base and large variants.

6.1.4 Transformer Output

After passing our encoded text through BERT, we have for each input document a contextually encoded representation of the question and document. We first construct a single set of question word embeddings by averaging the values across each encoding of the question. We then project the document and question embeddings from the dimensionality of the BERT model (768 for BERT-base-uncased) to that of the MAC model (512 for our base model) using a simple linear layer[1]. Finally, we extract the embedding produced for the '[CLS]' token (added in the text encoding step) for each document, and use this as a 'document embedding' - a single vector representation of each document. We also shrink these document embeddings to have dimensionality 512 with a single linear layer.

[1] A linear layer here is defined as $\text{Linear}(x) = W^{d \times b} \cdot x + b$, where W and b are learnable weight and bias parameters, d is the input dimensionality, and b is the output dimensionality.

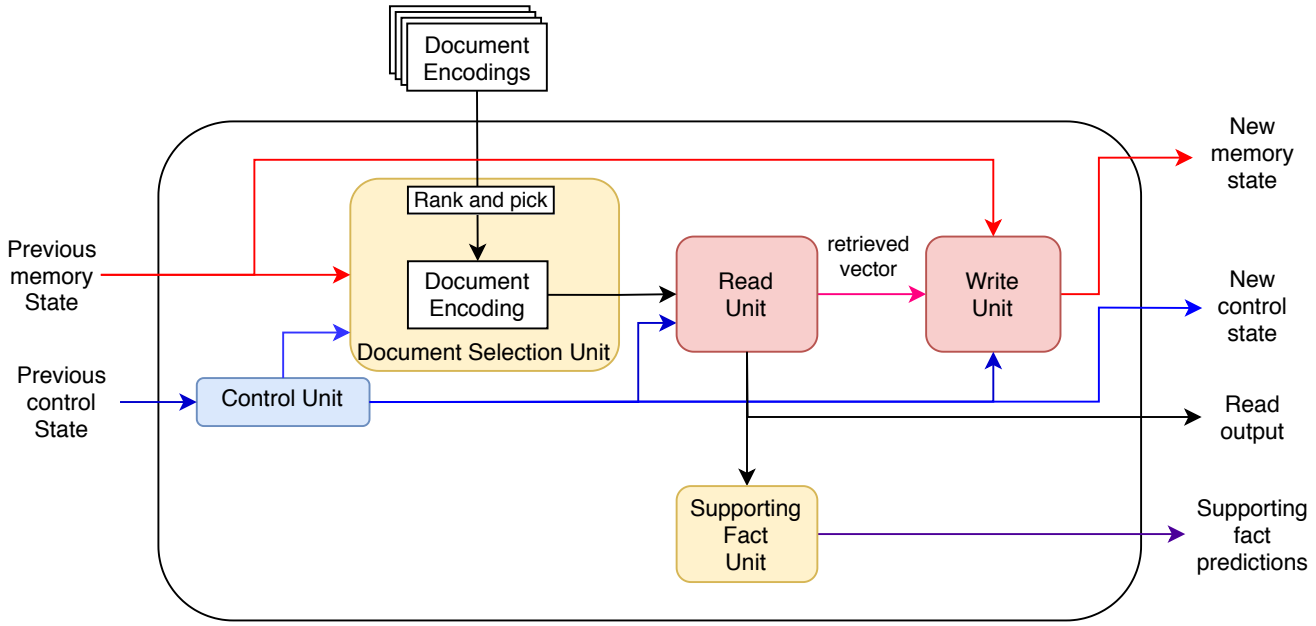


FIGURE 6.3. Our augmented MAC cell design for our BERT-based model.

6.1.5 Question Summary Vector

We swap out the LSTM-based method from the GloVe-based model with a self-attention mechanism (Zhong et al., 2019), used elsewhere in modular approaches to HotpotQA (Jiang and Bansal, 2019b). Removing the recurrent LSTM layer previously used for this provides a small speed-up when running our model, and empirically we found does not change the overall performance of the model. If we have question embeddings $[x_0, x_1, ...x_Q]$ (where Q is the number of tokens in the question), then the self-attention mechanism that produces question vector q is:

$$a_i = \text{softmax}(\tanh((W \cdot \tanh(x_i) + b))) \tag{6.20}$$

$$q = \sum_{i=0}^{i=Q} x_i * a_i \tag{6.21}$$

This question summary vector shares the same dimensionality as the shrunken word embeddings. This means that the question summary vector has dimensionality 512 in our base model.

6.2 MAC Cell

The design of our MAC cell remains similar to the design presented in chapter 4, with two notable augmentations: a document selection unit and a supporting fact unit. The document selection unit simply ranks the input documents and picks one for the cell to read, passing it to the read unit. The supporting fact unit simply uses the output from the read unit and a GRU layer to make supporting fact predictions for the chosen document. Other units in the MAC cell retain the exact same functionality as discussed in chapter 7. We always only use two cells, as HotpotQA is designed such that only two documents are used to answer any given question. We provide an overall diagram of this design in figure 6.3.

6.2.1 Document Selection Unit

The document selection unit is responsible for selecting the document to be passed to the rest of the MAC network. This is performed in a manner similar to Asai et al. (2020), using the control and memory states of the MAC cell to rank documents. We first project the document CLS embeddings, control state, and memory state using linear layers:

$$d'_i = W^{d\times d} d_i + b \tag{6.22}$$

$$c'_j = W^{d\times d} c_j + b \tag{6.23}$$

$$m'_{j-1} = W^{d\times d} m_{j-1} + b \tag{6.24}$$

Where d_i is the document embedding of the i^{th} document, d is hidden dimensionality of the MAC cell (512 for our base model). c_j and m_j represent the control and memory states of the j^{th} MAC cell respectively. Note we use the memory state output by the previous MAC cell here as the next memory state will be constructed from the document selected by this unit.

We then calculate interactions between each state and the document embeddings by performing element-wise multiplication and then concatenate these interactions together with the original embeddings. We pass this concatenated form through a linear layer and sigmoid function to produce a score for each document, representing the probability of selecting this document at this step.

$$cd_i = c'_j * d'_i \tag{6.25}$$

$$md_i = m'_j * d'_i \tag{6.26}$$

$$s_i = \sigma(W^{d\times 1}[d'_i; cd_i; md_i] + b) \tag{6.27}$$

We take the most likely document and pass it into the read unit, which operates as described in chapter 4.

6.2.2 Supporting Facts Unit

The supporting facts unit is responsible for predicting supporting facts from the output of the read unit, predicting which sentences support the final answer. This unit simply adapts the baseline method for supporting fact prediction: we pass the output from the read unit through a bidirectional GRU layer, and then use the output to construct sentence embeddings that are then scored to determine which are supporting facts.

$$out' = \text{GRU}(out) \tag{6.28}$$

$$s_i = [out'^f_i; out'^b_i] \tag{6.29}$$

$$s'_i = W^{d\times 1} s_i + b \tag{6.30}$$

$$\text{pred}_i = \sigma(s'_i) \tag{6.31}$$

Where out'^f and out'^b refer to the outputs from the forward and backwards GRUs respectively. We mark a sentence as a supporting sentence if pred_i is over 0.5 (although this threshold can be further tuned to maximise performance).

6.2.3 Other Unit Changes

In addition to the two units added to our design, we make some additional minor changes to the MAC network itself: we do not utilise the memory gate nor integrate the control state into the memory in the write unit as done in the GloVe-based model (see section 4.2.3 for details on these components).

6.3 Output Unit

Our output unit is similar to the GloVe-based model discussed in chapter 4, but made simpler. We first predict yes/no/span in an identical manner: we concatenate the question summary vector and final memory state, and then use a linear layer with a softmax activation function to make the prediction. Then, for span-based answering, we take the output from the last MAC cell and pass it through two linear layers to predict the start and end location of the answer in the chosen document, similar to how question answering is performed in a vanilla BERT model (Devlin et al., 2019).

$$s = W^{d \times 1} out + b \tag{6.32}$$

$$e = W^{d \times 1} out + b \tag{6.33}$$

$$s_{pred} = \text{argmax}(\text{softmax}(s)) \tag{6.34}$$

$$e_{pred} = \text{argmax}(\text{softmax}(e)) \tag{6.35}$$

Where out is the output from the final MAC unit and s_{pred} and e_{pred} are the predicted start and end locations of the answer respectively. By removing the recurrent GRU layers used in the GloVe-based model, we speed up the runtime of our model with no detected cost to performance (since BERT already provides incredibly strong processing capabilities). Similarly, removing the self-attention layer used in the GloVe-based model reduces the size of the model with no detected cost to performance.

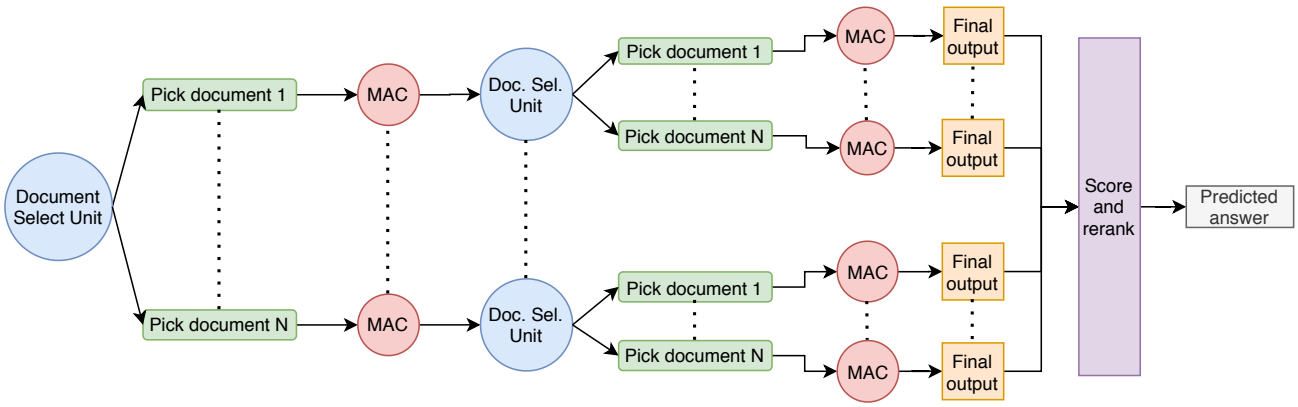


FIGURE 6.4. A diagram of how beam search operates in our model.

6.3.1 Beam Search

As the most likely document sequence may not match the most likely answer prediction, we investigate using beam search at inference time over the chosen documents. This means that at each document selection step in the MAC cell, we split the model's prediction process into k parallel paths (k is called the 'beam width'), where each path represents a different document being selected. After two MAC cells, we then output a result as above and predict the likelihood of the chosen documents by each cell being correct. This allows us to then rank each prediction process based on this final score, outputting the answer of the top-ranked process. We provide a diagram of this process in figure 6.4. By default, we do not use this mechanism (i.e. by default, we use a beam width of 1).

The score for each prediction process is calculated in the output unit by passing the concatenation of the final memory state and question vector through a two-layer multi-layer perceptron.

$$\text{score} = W^{d\times 1}(\text{elu}(W^{d\times d}qm))) \tag{6.36}$$

Where qm is the concatenated question and memory vectors, and 'elu' is the ELU activation function (Clevert et al., 2016).

6.4 Loss and Training

We utilise the same loss functions as the GloVe-based model, with two additions. First, we calculate the binary cross-entropy loss for the document predictions made by the document selection unit. Secondly, we use the reranking loss from Burges et al. (2005) for training the reranking component when using beam search:

$$L_{rank}(s_i, s_j) = [y_i - \sigma(s_i, s_j)]^2 \tag{6.37}$$

Where s_i and s_j are two reranking scores calculated by the model from two different documents, and $y_i = 1$ if s_i is the score for a correct document path[2], and 0 otherwise. During training, we force our

[2]i.e. a path that ends on the correct answer document after going through the other supporting document. The correct answer document is always the one that contains the answer text, and if both documents contain the answer text, we determine

model to take particular document paths (i.e. overriding its next document predictions). We generate scores for three paths: the correct path, the correct path but flipped, and a randomly-generated incorrect path (where the final document is not the answer document). We then calculate the reranking loss between the correct path score and the two incorrect path scores. If the flipped path is still correct (e.g. when the answer appears in both supporting documents, or when the answer is yes/no), we simply mask the loss between it and the 'correct' path to avoid confusing our model. Apart from the reranking loss, all other losses are only calculated when using the 'correct path'.

6.5 Optimisation

We use the Adam optimiser (Kingma and Ba, 2015) with a learning rate of $5e-5$ and halve the learning rate when the answer F1 score does not improve. We train the model for 4 epochs, which we empirically found to always be enough for the model to achieve optimal performance.

6.6 Conclusion

In this section, we have introduced our BERT-based model in detail. This model utilises a novel document selection unit to iteratively choose and read documents, integrating the reading and document selection steps often performed separately in other models. By iteratively reading documents and writing important facts to its hidden memory state, the MAC cell is able to better select documents and determine the answer to a given question. Furthermore, this model can easily be trained end-to-end by training on automatically detected correct document paths during training. We also introduce a beam search mechanism that allows our model to investigate multiple document paths at inference time if necessary. This approach thus merges the strengths of BERT and MAC into one model.

if the title of one document appears in the other. If so, the document whose title appears in the other is set as the answer document. If not, we set one of the documents randomly as the answer document. We similarly set a random order for yes/no questions, as this answer can be output independent of chosen document.

CHAPTER 7

BERT-based Model Results

In this chapter, we explore the performance and behaviour of our document selection-focused BERT-based model design. We perform similar analyses as those performed for our GloVe-based model, with some differences.

7.1 Quantitative Evaluation

7.1.1 Performance

We first evaluate the performance of our document selection-focused MAC model and compare it against other BERT-based models. Similar to our GloVe-based approach, we compare both just using the dev set and using a test set split from the original dev set (which is completely held out during training). We compare against other models on the dev set in table 7.1, and against other models on the test set in table 7.2.

As we can see, while our design is slightly below the performance of the SAE and HGN models, it is competitive with the recurrent retriever, which utilises a similar document selection method but with a

	Answer		Supporting Facts		Joint	
	EM	F1	EM	F1	EM	F1
Doc. Sel. MAC (ours)	59.04	73.38	58.49	85.18	39.00	65.34
BERT baseline	60.08	74.26	**60.32**	86.28	**40.30**	66.61
RR	59.40	73.30	57.40	84.60	-	-
SAE	**61.32**	**74.81**	58.06	85.27	39.89	66.45
HGN	-	74.76	-	**86.61**	-	**66.90**

TABLE 7.1. Performance comparison of document selection MAC with other BERT-based models on the HotpotQA distractor dev set. RR refers to the recurrent retriever model (Asai et al., 2020) referred to in chapter 3. Dash indicates scores unavailable or not reported.

	Answer		Supporting Facts		Joint	
	EM	F1	EM	F1	EM	F1
Doc. Sel. MAC (ours)	58.71	72.68	**58.82**	**85.27**	**39.37**	64.85
BERT baseline	-	-	-	-	-	-
SAE	**60.36**	**73.58**	56.93	84.63	38.81	**64.96**

TABLE 7.2. Performance comparison of document selection MAC with other BERT-based models on HotpotQA distractor test sets. Note that the HGN and RR models do not report BERT-based scores on the HotpotQA distractor test set. Test set used for SAE is the official HotpotQA distractor test set.

	Answer		Supporting Facts		Joint	
	EM	F1	EM	F1	EM	F1
Doc. Sel. MAC, K = 1	**59.04**	**73.38**	**58.49**	**85.18**	**39.00**	**65.34**
- hidden state doc. sel.	36.19	47.13	0.00	58.24	0.00	28.78
- control state	58.53	72.92	58.15	85.44	38.49	64.97
- control biattn	58.81	72.92	58.46	85.21	38.99	64.90
- memory in read	58.64	73.00	58.46	85.22	38.57	64.99
+ beam search, K = 5	58.62	73.08	56.79	84.41	37.85	64.61
+ beam search, K = 10	57.30	71.63	47.41	80.30	32.38	60.82

TABLE 7.3. Ablation results on HotpotQA distractor dev set for document selection based MAC. See section 7.1.2 for details on each ablation. K refers to beam width for the beam search component.

dedicated separate reader model. This shows the efficacy of document selection: our document selection design is able to achieve results competitive with a model with a dedicated BERT reader module. Furthermore, we see that our document selection model is still able to achieve supporting fact scores on par with the SAE model, again showing the complex graph methods used by the SAE model are largely unnecessary compared to our simple GRU-based approach.

Note that we do not test single-hop datasets with this model, since there is no document selection process required for such datasets. As the majority of the modifications made by our model are for better document selection, this means on the single hop setting our model is roughly equivalent to a baseline BERT model.

7.1.2 Ablations

Next, we investigate the effect of various components of our design with the ablations shown in table 7.1.2. The ablations we test are:

- **-hidden state doc. sel.**: We remove the use of the memory and control state for document selection, and instead just directly use the CLS tokens from the BERT model encoding for ranking at each step.
- **-control state**: We remove the use of the control state in the document selection and write unit and use of the control-based bi-attention in the read unit.
- **-biattn**: We remove the control-based bi-attention layer in the read unit.
- **-memory in read**: We remove the integration of the memory vector with the knowledge base in the read unit. This effectively means our answer span prediction is performed without knowledge of prior documents in the final cell.
- **+ beam search, K = 5**: We utilise the beam search component described in section 6.3.1, with a beam width of 5.
- **+ beam search, K = 10**: We utilise the beam search component with a beam width of 10.

Based on these ablations, we can see that the beam search component only hurts the overall performance of the model, despite the effectiveness of beam search-based reranking in open-domain QA (Asai et al., 2020). We believe this is due to the distractor documents in the distractor setting being easy enough to distinguish from others, given they are chosen using just TF-IDF overlap (Yang et al., 2018), rather than specifically being chosen to distract the powerful BERT model. With this, the model is generally able to distinguish the correct supporting documents most of the time. However, we can see the hidden memory state is vital in our model for better document selection and higher performance. This is likely due to the information uncovered in previous documents aiding the document selection process in the case of bridge questions, which are explicitly crafted for this sort of reasoning. Finally, we see that the control state and bi-attention provide only small benefits, indicating the model is largely not utilising the control unit, similar to the GloVe-based model.

7.1.3 Parameter Tuning

7.2 Qualitative Evaluation Results

We now examine the behaviour of this BERT-based model, applying similar analyses as those performed for the GloVe-based model. These provide further insight into the performance of the model and expose future directions for research on multi-hop QA.

# Samples	Bridge			Comparison (np)			Comparison (p)		
	5918			1029			458		
	F1	SP F1	J F1	F1	SP F1	J F1	F1	SP F1	J F1
BERT Baseline	**74.86**	**85.30**	**66.59**	**68.96**	89.59	**63.87**	78.38	91.47	72.93
Doc. Sel. MAC	73.65	83.95	64.93	66.83	**90.42**	62.16	**81.00**	**93.89**	**76.74**

TABLE 7.4. Performance of BERT-based models broken down by question type on HotpotQA distractor dev set. 'np' and 'p' stand for 'non-polar' and 'polar' respectively. 'F1', 'SP F1', and 'J F1' refer to answer F1, supporting fact F1, and joint F1 respectively.

7.2.1 Sample Breakdown

We first examine the performance of our BERT-based design on the different question types in HotpotQA in table 7.4. Interestingly, while the document selection model is unable to achieve above baseline performance in bridge questions, it is much stronger at comparison questions and especially strong at polar comparison questions. This is likely because polar comparison questions do not require proper document selection to find the answer (since the model has a dedicated yes/no/span prediction module). Hence, the difficulty of selecting the correct answer document does not result in a decrease in a performance for these questions. We also hypothesise that the direct use of memory vector for these question types provides stronger guidance on the memory state of the MAC cells, and thus on what information the MAC cells should be reading from each document. This would explain the relatively poor performance of our model on non-polar comparison questions, for which the model still has to predict the answer location, meaning the memory vector is only indirectly trained to carry information required to locate the final answer. The reduced performance on bridge questions is likely due both to the difficulty of selecting the correct answer-containing document (Wang et al., 2019c) and the small reduction in performance that occurs when not jointly encoding documents that we showed in section 5.3.2.

Next, we examine the performance of our document selection model by examining its performance across different answer types in table 7.5. Notably, the model performs quite well across a set of different answer types but does not perform well for the most common answer type: a person's name ('person'). This suggests that the model is still able to reason about different types of entities quite well, but struggles to achieve better performance for some of the more common answer types in HotpotQA.

Examining the types of errors made by our document selection models in table 7.6, we see that our model suffers more from document selection errors, in which the model has failed to find the correct answer document. This suggests that improving the document selection mechanism of our model could

	# Questions	BERT Baseline	Doc. Sel. MAC
Artwork	16	93.75	**100.00**
Date	27	**85.77**	84.66
Adjective	9	70.27	**81.20**
Yes/No	21	76.19	**80.95**
Group	27	74.94	**79.26**
Location	54	70.48	**74.82**
Number	23	**74.74**	72.90
Person	73	**76.91**	72.74
Proper noun	27	**74.57**	65.93
Event	6	**67.04**	64.54
Common noun	15	38.09	**53.08**
Mislabel	2	**5.56**	**5.56**

TABLE 7.5. Answer F1 for different answer types for our BERT-based models, based on a sample of 300 questions from HotpotQA distractor dev set.

	BERT Baseline	Doc. Sel. MAC
Actually Correct	18	18
Commonsense	4	4
Discrete Reasoning	13	7
Mislabel	7	5
General	23	14
Multi-hop	18	22
Incorrect Doc. Sel.	17	30

TABLE 7.6. Number of errors made by BERT-based model from a sample of 100 errors. See Appendix A for details on the error types.

potentially lead to vastly improved performance. Furthermore, the greater proportion of multi-hop errors further suggests that the model is unable to integrate information from across different documents well. This gives further evidence that our model is unable to make good use of the memory vector for communicating information across documents.

Finally, we examine the accuracy of our document selection module. Unlike previous approaches, which choose a set of documents and then allow a reader model access to the entire set at once, our model is order-sensitive: the document chosen by the final cell must contain the answer (if the answer is not yes/no), as otherwise it will be impossible for the model to predict the correct answer (since it simply selects text from the final document as its answer). As such, we examine not just the accuracy of the model in selecting the two supporting documents, but also its accuracy in selecting the correct answer document. As seen across tables 7.7 and 7.8, our model is almost perfect at selecting documents for comparison questions, but struggles more with bridge questions, and particularly struggles at finding the

	Bridge	Bridge (both)	Comp (p)	Comp (np)	**All**
Any 1 document correct (unordered)	99.57	99.46	100.00	99.90	99.62
Both documents correct (unordered)	88.26	86.82	99.56	98.45	90.09
First document correct (ordered)	90.16	-	-	71.82	82.42
Second document correct (ordered)	88.62	-	-	71.91	81.22
Both documents correct (ordered)	84.43	-	-	71.53	77.93

TABLE 7.7. Percentage of correct selected documents split by question type as chosen by the document selection MAC. 'Bridge (both)' refers to bridge questions which have the answer text in both supporting documents. Dashes indicate scores that would not make sense to record, as those question types allow any document order.

	Bridge	Bridge (both)	Comp (p)	Comp (np)	**All**
Any 1 document correct (unordered)	99.73	99.80	99.13	99.42	99.66
Both documents correct (unordered)	91.06	90.60	98.25	97.18	92.26

TABLE 7.8. Percentage of correct selected documents split by question type as chosen by the SAE document selection method. 'Bridge (both)' refers to bridge questions which have the answer text in both supporting documents.

correct answer document for non-polar comparison questions. These results largely match our intuition: comparison questions are easier to select documents for as the relevant entities are usually directly named in the question, while bridge questions often require selecting answer documents which are non-obvious when looking just at the question. Furthermore, the low document selection performance for non-polar comparison questions is due to the fact that selecting the correct document for these questions is tantamount to choosing the correct answer, since each potential answer is in a different document (and so choosing a document means the model must choose that potential answer as its predicted answer). Comparing our method to the SAE method, we note that we achieve better performance on selecting comparison documents, but far worse on selecting bridge documents. We believe this performance gap is due to the more sophisticated training mechanism of the SAE, and its use of a more powerful BERT model (BERT-WWM-large as compared to BERT-base[1]). Thus our document selection method is clearly promising, but requires further work.

7.2.2 Attention Maps

Finally, we examine the attention maps produced by each MAC cell. As the document selection MAC only processes one cell at a time, visualising the attention maps produced by the read and control units is much simpler, involving less text. We find that these attention maps are surprisingly interpretable

[1]Tu et al. (2020) do not mention using this model, but the code provided with their paper uses this larger BERT model.

in the case of bridge questions, clearly highlighting the bridge entity, although the control attention is less interpretable, as seen below. Note that the examples below utilise BERT-base-uncased tokenisation, which converts all text to lowercase before tokenising. As this tokenisation scheme breaks words up into sub-parts, we prepend '##' to sub-parts of words to indicate they are not the start of a new word.

Question ID 5abd94525542992ac4f382d2

Cell #1

Control: 2014 s / s is the debut album of a south korean boy group that was formed by who ?

Read: < t > 2014 s / s < / t > 2014 s / s is the debut album of south korean group winner . it was released on august 12 , 2014 by the group ' s record label , y ##g entertainment . the members were credited for writing the lyrics and composing the majority of the album ' s songs .

Cell #2

Control: 2014 s / s is the debut album of a south korean boy group that was formed by who ?

Read: < t > winner (band) < / t > winner (hangul : [UNK]) , often stylized as winner , is a south korean boy group formed in 2013 by y ##g entertainment and debuted in 2014 . it currently consists of four members , jin ##wo ##o , se ##ung ##ho ##on , min ##o and se ##ung ##yo ##on . originally a five - piece group with tae ##hy ##un , who later departed from the group in november 2016 .

In this example, it is clear when looking at the read unit attention the model first focuses on finding the band name ('Winner') before then focusing on the formation of the band in the second article. However, the control unit attention does not seem to follow, focussing on the album name in the second cell, where this information is not useful for finding the answer. We find that other bridge questions similarly show this pattern of more interpretable read unit attention, but less useful control attention[2]. In contrast, comparison questions show little in the way of interpretable reasoning processes in both their control and read attention maps:

[2]We provide further attention maps from the HotpotQA dev set in Appendix C.

Question ID 5a8b57f25542995d1e6f1371

Cell #1

Control: were scott derrick ##son and ed wood of the same nationality ?

Read: < t > scott derrick ##son < / t > scott derrick ##son (born july 16 , 1966) is an american director , screenwriter and producer . he lives in los angeles , california . he is best known for directing horror films such as " sinister " , " the ex ##or ##cis ##m of emily rose " , and " deliver us from evil " , as well as the 2016 marvel cinematic universe installment , " doctor strange . "

Cell #2

Control: were scott derrick ##son and ed wood of the same nationality ?

Read: < t > ed wood < / t > edward davis wood jr . (october 10 , 1924 – december 10 , 1978) was an american filmmaker , actor , writer , producer , and director .

As we can see in the above example, the control unit attention shifts little across the two cells, and despite the question asking about nationality, the nationalities of the two filmmakers have no attention paid to them in the read unit. This is likely due to the powerful pre-processing of BERT: as it creates contextual word embeddings, it is highly likely that the information about the nationalities of the two filmmakers has already been spread and integrated into the representations of other tokens within the text. Furthermore, as the final answer text is the name of one of the filmmakers, the model itself learns during training to pay more attention to the names, rather than the attributes being examined to answer the question. This highlights the perils of using BERT: while we can get some seemingly fairly interpretable attention maps, the large size of BERT means that it can effectively shift information to any token it pleases with minimal effect on the answer predictions we make. The very nature of contextual representations is what allows this: by allowing words to contain information about their context, we naturally lose the ability to pinpoint the exact contribution that singular word has on an answer. However, the fact that the read unit of our document selection model highlights bridge entities without being trained to suggests that in this case it is exposing some information about the reasoning process underlying the model, and so still can aid a user in determining why the model has made a particular prediction.

Thus, while the attention maps of the document selection model still leave something to be desired, we have seen that they are still quite promising, with interpretable qualities arising without being specifically trained into the model. Furthermore, the document selection process and 'one cell, one document' paradigm makes visualising these attention maps far simpler than in the case of the GloVe-based models, where we had to visualise large amounts of text at once.

7.3 Conclusion

In this chapter, we have thoroughly explored the performance and behaviour of our BERT-based, document selection-focused model design. While this model is not quite state-of-the-art, it performs incredibly well on non-polar comparison questions and provides interpretable attention maps which expose its underlying reasoning process. Furthermore, our model slightly outperforms a strong existing 'read and retrieve' baseline, the recurrent retriever. Thus, while our model has some shortcomings, we believe it is an effective and interesting design, with several clear potential future research directions for improving its performance and interpretability.

CHAPTER 8

Conclusion

8.1 Future Work

In this work, we have explored the problem of multi-hop QA by utilising MAC cells, a novel design originally applied to visual question answering that carries a strong inductive bias for multi-hop reasoning. While we have focused on an extractive setup, where the model must extract an answer from input text, future work could examine multiple-choice setups such as WikiHop (Welbl et al., 2018), which require slightly different answer mechanisms but are still compatible with MAC cell-based approaches. Furthermore, we have not thoroughly investigated the utility of graph-based representations of text, which have proven effective in certain scenarios (De Cao et al., 2019; Fang et al., 2020). This shows there are two clear avenues for further research with our model design: either through expanding the design to utilise different QA designs or through applying the design to other tasks and setups.

In addition, we have shown that the distractor setting of HotpotQA relies heavily on document choice, rather than specific reader components. This suggests researchers should control for document selection methods in future work, since it is currently unclear whether strong approaches are strong due to their document selection or document reading model. This suggests that the distractor models have much to learn from the full-wiki setting of HotpotQA, where document selection is often the main problem being examined. Future work could also examine creating stronger QA datasets for which document selection is less effective - for example, ensuring documents have multiple potential answers that can only be disambiguated by referring to a secondary (or more) document(s).

Finally, we note that the interpretable aspects of the MAC cell are less effective in the context of text-only tasks, where contextual representations and large inputs make effectively identifying which core words are contributing to a particular answer more difficult than in computer vision tasks. Future work could examine how to improve this interpretability, which can potentially be achieved via further supervision

during training, restricting modules to communicate only via interpretable text, or better integration of supporting facts supervision into the cell design itself. Further examination of the information written to the memory and control states via the use of probing tasks or toy datasets would also aid in uncovering how our MAC networks reason, and aid in measuring how faithful the attention maps produced in the read and control units are to the actual underlying reasoning process of our networks.

Thus, while we have shown that MAC cell-based designs are promising for multi-hop QA, there are still many potential areas for improvement and research on these models.

8.2 Contributions and Conclusion

We conclude by summarising the four major contributions we have made in this work:

(1) We adapted the MAC network to machine reading comprehension, bringing a popular image-based model to a text-based task.
(2) We showed this adapted model provides strong performance compared to existing modular approaches, whilst also being more interpretable and applicable to non-multi-hop datasets with minimal loss in performance.
(3) We showed that good performance on HotpotQA, a highly popular multi-hop dataset, revolves around good document selection methods, which are largely under-examined in the distractor setting.
(4) We designed a multi-hop model that can achieve competitive performance on HotpotQA primarily through its document selection. This model joins the strengths of the MAC and BERT models, using MAC cells to pick and read documents, and a BERT model to produce rich contextual representations of the question and input documents.

In this work, we have thus provided a thorough exploration of MAC cells and multi-hop reasoning, showcasing the multiple ways MAC cells are useful for multi-hop QA. We have shown that MAC cells carry a strong inductive bias for multi-hop reasoning and can outperform existing modular approaches on HotpotQA, while also generalising to adversarial and single-hop QA datasets. In addition, we have shown that current approaches to the HotpotQA distractor setting largely on a relatively under-examined document selection step. Finally, we designed a competitive that jointly selects and reads documents, combining MAC cells with a BERT model in an interpretable manner. We hope this work inspires

further research into designing multi-hop datasets that require strong cross-document reasoning and further research into applying modular networks to text-based tasks.

Bibliography

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization.

Petr Baudiš and Jan Šedivý. 2015. Modeling of the question answering task in the yodaqa system. In Josanne Mothe, Jacques Savoy, Jaap Kamps, Karen Pinel-Sauvagnat, Gareth Jones, Eric San Juan, Linda Capellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 222–228. Springer International Publishing, Cham.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544. Association for Computational Linguistics, Seattle, Washington, USA.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217. Association for Computational Linguistics, Barcelona, Spain.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 89–96. Association for Computing Machinery, New York, NY, USA.

Eugene Charniak, Yasemin Altun, Rodrigo de Salvo Braz, Benjamin Garrett, Margaret Kosmala, Tomer Moscovich, Lixin Pang, Changhee Pyo, Ye Sun, Wei Wy, Zhongfa Yang, Shawn Zeiler, and Lisa Zorn. 2000. Reading comprehension programs in a statistical-language-processing class. In *ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879. Association for Computational Linguistics, Vancouver, Canada.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111. Association for Computational Linguistics, Doha, Qatar.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 845–855. Association for Computational Linguistics, Melbourne, Australia.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (elus). In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.

Yiming Cui, Ting Liu, Zhipeng Chen, Shijin Wang, and Guoping Hu. 2016. Consensus attention-based neural networks for Chinese reading comprehension. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1777–1786. The COLING 2016 Organizing Committee, Osaka, Japan.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. 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)*, pages 2306–2317. Association for Computational Linguistics, Minneapolis, Minnesota.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistic (NAACL)*, pages 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota.

Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*.

Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W. Cohen. 2020. Differentiable reasoning over a virtual knowledge base. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2694–2703. Association for Computational Linguistics, Florence, Italy.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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)*, pages 2368–2378. Association for Computational Linguistics, Minneapolis, Minnesota.

M. Dunn, Levent Sagun, M. Higgins, V. U. Güney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *ArXiv*, abs/1704.05179.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Yair Feldman and Ran El-Yaniv. 2019. Multi-hop paragraph retrieval for open-domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2309. Association for Computational Linguistics, Florence, Italy.

David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. 2010. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79.

Jinlan Fu, Yi Li, Qi Zhang, Qinzhuo Wu, Renfeng Ma, Xuanjing Huang, and Yu-Gang Jiang. 2020. Recurrent memory reasoning network for expert finding in community question answering. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 187–195. Association for Computing Machinery, New York, NY, USA.

Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: An automatic question-answerer. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '61 (Western), page 219–224. Association for Computing Machinery, New York, NY, USA.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2020. Neural module networks for reasoning over text. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Sam Gustin. 2017. Watson supercomputer terminates humans in first jeopardy round. *Wired*. Accessed 6/11/2020.

T. J. Hazen. 2019. Bringing the power of machine reading comprehension to specialized documents. *Microsoft Research*. Accessed 6/11/2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 1693–1701. Curran Associates, Inc.

Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. 1999. Deep read: A reading comprehension system. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 325–332. Association for Computational Linguistics, College Park, Maryland, USA.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Drew A Hudson and Christopher D Manning. 2018. Compositional attention networks for machine reasoning. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Yichen Jiang and Mohit Bansal. 2019a. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2726–2736. Association for Computational Linguistics, Florence, Italy.

Yichen Jiang and Mohit Bansal. 2019b. Self-assembling modular networks for interpretable multi-hop reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4474–4484. Association for Computational Linguistics, Hong Kong, China.

Di Jin, Shuyang Gao, Jiun-Yu Kao, Tagyoung Chung, and Dilek Hakkani-Tür. 2020. MMM: multi-stage multi-task learning for multi-choice reading comprehension. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8010–8017. AAAI Press.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611. Association for Computational Linguistics, Vancouver, Canada.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA.

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 908–918. Association for Computational Linguistics, Berlin, Germany.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262. Association for Computational Linguistics, New Orleans, Louisiana.

J. Kiefer and J. Wolfowitz. 1952. Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.*, 23(3):462–466.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Guillaume Le Berre and Philippe Langlais. 2020. Attending knowledge facts with bert-like models in question-answering: Disappointing results and some explanations. In Cyril Goutte and Xiaodan Zhu, editors, *Advances in Artificial Intelligence*, pages 356–367. Springer International Publishing, Cham.

Dongjun Lee, Sohee Yang, and Minjeong Kim. 2019. Claf: Open-source clova language framework. `https://github.com/naver/claf`.

Wendy Grace Lehnert. 1977. *The Process of Question Answering.* Ph.D. thesis, USA.

Zachary C. Lipton and Jacob Steinhardt. 2019. Research for practice: troubling trends in machine-learning scholarship. *Commun. ACM*, 62(6):45–53.

Tie-Yan Liu. 2011. *Learning to Rank for Information Retrieval*. Springer Berlin Heidelberg, Berlin, Heidelberg.

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

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

John Markoff. 2011. Computer wins on 'jeopardy!': Trivial, it's not. *The New York Times*. Accessed 6/11/2020.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391. Association for Computational Linguistics, Brussels, Belgium.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409. Association for Computational Linguistics, Austin, Texas.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257. Association for Computational Linguistics, Florence, Italy.

Timo Möller, Anthony Reina, Raghavan Jayakumar, and Lawrence Livermore. 2020. Covid-qa: A question & answering dataset for covid-19.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814. Omnipress, Madison, WI, USA.

Pandu Nayak. 2019. Understanding searches better than ever before. *Google*. Accessed 6/11/2020.

Hwee Tou Ng, Leong Hwee Teo, and Jennifer Lai Pheng Kwan. 2000. A machine learning approach to answering questions for reading comprehension tests. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 124–132. Association for Computational Linguistics, Hong Kong, China.

Xiaoman Pan, Kai Sun, Dian Yu, Jianshu Chen, Heng Ji, Claire Cardie, and Dong Yu. 2019. Improving question answering with external knowledge. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 27–37. Association for Computational Linguistics, Hong Kong, China.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, Doha, Qatar.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150. Association for Computational Linguistics, Florence, Italy.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789. Association for Computational Linguistics, Melbourne, Australia.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics, Austin, Texas.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203. Association for Computational Linguistics, Seattle, Washington, USA.

Ellen Riloff and Michael Thelen. 2000. A rule-based question answering system for reading comprehension tests. In *ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

Nan Shao, Yiming Cui, Ting Liu, S. Wang, and G. Hu. 2020. Is graph structure necessary for multi-hop reasoning? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Dominican Republic.

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasonet: Learning to stop reading in machine comprehension. KDD '17, page 1047–1055. Association for Computing Machinery, New York, NY, USA.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4515. Association for Computational Linguistics, Hong Kong, China.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601. Association for Computational Linguistics, Florence, Italy.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020*, pages 9073–9080. AAAI Press.

A. M. Turing. 1950. Computing machinery and intelligence. *Mind*, 59(236):433–460.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 3266–3280. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*.

Haoyu Wang, Mo Yu, Xiaoxiao Guo, Rajarshi Das, Wenhan Xiong, and Tian Gao. 2019c. Do multi-hop readers dream of reasoning chains? In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 91–97. Association for Computational Linguistics, Hong Kong, China.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

William A. Woods. 1977. Lunar rocks in natural English: Explorations in natural language question answering. In Antonio Zampolli, editor, *Linguistic Structures Processing*, pages 521–569. North Holland, Amsterdam.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763. Curran Associates, Inc.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380. Association for Computational Linguistics, Brussels, Belgium.

Wenpeng Yin, Sebastian Ebert, and Hinrich Schütze. 2016. Attention-based convolutional neural network for machine comprehension. In *Proceedings of the Workshop on Human-Computer Question Answering*, pages 15–21. Association for Computational Linguistics, San Diego, California.

Jianxing Yu, Zhengjun Zha, and Jian Yin. 2019. Inferential machine comprehension: Answering questions by recursively deducing the evidence chain from text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2241–2251. Association for Computational Linguistics, Florence, Italy.

Victor Zhong, Caiming Xiong, Nitish Shirish Keskar, and Richard Socher. 2019. Coarse-grain fine-grain coattention network for multi-evidence question answering. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*.

APPENDIX A

Error Types

In this section we provide descriptions of the error types we use to classify errors across this work (particularly tables 5.8 and 7.6).

- **Actually Correct**: The answer proposed by the model is correct, although it does not match the ground-truth label. This can occur when the predicted answer is a different grammatical form (e.g. singular vs plural), or simply a different answer that is still correct (in the case when a question has multiple potential answers).
- **Commonsense**: The answer can only be found using knowledge external to provided documents. For example, some questions in HotpotQA require prior knowledge that certain US states are north of or south of other states.
- **Discrete Reasoning**: The answer can only be found using discrete reasoning, which includes counting and number comparison. This includes questions such as 'Who is older?' or 'Who has more albums?'.
- **Multi-hop**: The model has failed to perform multi-hop reasoning, which is indicated by it either predicting the bridge entity as the answer instead of the actual answer, or by following a wrong (but plausible) bridge entity to a wrong answer. This also includes cases when comparisons are incorrectly made, as this indicates the model has failed to find and compare disjoint facts.
- **Superspan**: The model has predicted a span of text containing the correct answer. This generally only occurs when the model is unable to produce the exact answer due to incorrect tokenisation.
- **No Answer**: The model produced no answer for this question. For the GloVe-based model this only occurs when the input text is too large to process (over 2250 tokens, in our case).
- **Mislabel**: The ground truth label is incorrect, and so no plausible answer will be marked as correct in the F1 or EM metrics.

- **General**: A general error which does not fit in an above group. This is a catch-all category, and usually indicates the model has genuinely misunderstood either the question or underlying text in some way.
- **Incorrect Doc. Sel.**: The selected documents were incorrect, meaning the model does not have enough information required to predict the correct answer (and in most cases this means the answer text itself is not present in the selected documents).

APPENDIX B

GloVe-based Model Attention Maps

In this appendix we provide full attention maps of 4 sample questions from our 2-cell GloVe-based model. Colour indicates attention values, with darker red indicating higher attention values. Questions are referred to by their HotpotQA ID.

B.1 Maps for Question 5abd94525542992ac4f382d2

Cell #1

Question: 2014 S / S is the debut album of a South Korean boy group that was formed by who ?
Documents: <t> List of awards and nominations received by Shinee </t> South Korean boy group Shinee have received several awards and nominations for their music work . The group was formed by S.M. Entertainment in 2008 and released their first full - length album , " The Shinee World " , on August 28 , 2008 , which won the Newcomer Album of the Year at the 23rd Golden Disk Awards . The first single released from the album was " Sanso Gateun Neo (Love Like Oxygen) " and won first place on " M Countdown " on September 18 , 2008 making it the group 's first win on Korean music shows since debut . Their second album " Lucifer " (2010) produced two singles , " Lucifer " and " Hello " . For their outstanding choreography the group was nominated for the Best Dance Performance Award at the Mnet Asian Music Awards in 2010 . " Lucifer " also won the Disk Bonsang Award at the 25th Golden Disk Awards as well as the Popularity Award . On March 21 , 2012 the group released their fourth EP " Sherlock " for which the group was awarded another Disk Bonsang Award at the 27th Golden Disc Awards and the Bonsang Award at the 22nd Seoul Music Award . Also following the success of the lead single it was also nominated for Song of the Year at the 2012 Mnet Asian Music Awards . <t> Cho Kyuhyun </t> Cho Kyu - hyun (born February 3 , 1988) , better known mononymously as

Kyuhyun , is a South Korean singer and musical theatre actor . He is best known as a member of South Korean boy group Super Junior , its sub - groups Super Junior - K.R.Y. , Super Junior - M and a former member of the South Korean ballad group S.M. the Ballad . He is one of the first four Korean artists to appear on Chinese postage stamps . <t> 2014 S / S </t> 2014 S / S is the debut album of South Korean group WINNER . It was released on August 12 , 2014 by the group 's record label , YG Entertainment . The members were credited for writing the lyrics and composing the majority of the album 's songs . <t> History (band) </t> History (Korean : 히스토리) was a South Korean boy group formed by LOEN Entertainment in 2013 . They debuted on April 26 , 2013 with " Dreamer " , featuring the narration of their labelmate IU . They were LOEN Entertainment 's first boy group . They officially disbanded on May 12 , 2017 . <t> Winner (band) </t> Winner (Hangul : 위너) , often stylized as WINNER , is a South Korean boy group formed in 2013 by YG Entertainment and debuted in 2014 . It currently consists of four members , Jinwoo , Seunghoon , Mino and Seungyoon . Originally a five - piece group with Taehyun , who later departed from the group in November 2016 . <t> Madtown </t> Madtown (Hangul : 매드타운) , often stylized as MADTOWN , is a South Korean boy group formed in 2014 by J. Tune Camp . The group consists of Moos , Daewon , Lee Geon , Jota , Heo Jun , Buffy and H.O. Their debut album , " Mad Town " , was released on October 6 , 2014 . Two of the members , Moos and Buffy , originally debuted as the hip hop duo " Pro C " in 2013 . Madtown 's official fan - base name is Mad - people . Starting December 22 , 2016 , MADTOWN 's contract was sold to GNI Entertainment after J. Tune Camp closed . <t> List of songs written by Ravi </t> Ravi is a South Korean rapper , songwriter and producer , signed under Jellyfish Entertainment . He began his career as a rapper in 2012 in the South Korean boy group VIXX , and later formed VIXX 's first sub - unit VIXX LR with band mate Leo in 2015 . Ravi 's songwriting career began with his participation in co - writing VIXX 's debut single " Super Hero " . As of November 2016 with the release of " VIXX 2016 Conception Ker " , Ravi has contributed to the writing and composing of over 46 songs recorded by VIXX . Ravi is widely known for his participation of composing and songwriting rap portions for the group as well as lyrics and music . <t> SF9 (band) </t> SF9 (Korean : 에스에프나인 ; shortened from Sensational Feeling 9) is a South Korean boy group formed by FNC Entertainment . SF9

is the company 's first dance boy group to ever debut . SF9 debuted on October 5 , 2016 with the release of their first single album " Feeling Sensation " . <t> Seventeen discography </t> This is the discography of South Korean boy group Seventeen . Seventeen (Hangul : 세븐틴) , also stylized as SEVENTEEN or SVT , is a South Korean boy group formed by Pledis Entertainment in 2015 . They have released one album and four EPs . <t> BTS discography </t> The following is the discography of South Korean boy group BTS . The group debuted in South Korea on June 2013 with single album , " 2 Cool 4 Skool " , at number 5 on South Korean Week 31 Gaon Weekly Chart . They made a comeback on September 2013 with an extended play , " O!RUL8,2 ? " , which peaked at number 4 on Week 38 Gaon Weekly Chart . BTS then released their second extended play , " Skool Luv Affair " , in February 2014 , where it charted at number 1 on Week 18 Gaon Weekly Chart . This also marked the first time their album charted on international charts , Billboard World Albums and Japan 's Oricon Chart , specifically . A repackaged version of the album , " Skool Luv Affair Special Addition " which was released in May 2014 , also peaked at number 1 on Week 21 Gaon Weekly Chart .

Cell #2

Question: 2014 S / S is the debut album of a South Korean boy group that was formed by who ?

Documents: <t> List of awards and nominations received by Shinee </t> South Korean boy group Shinee have received several awards and nominations for their music work . The group was formed by S.M. Entertainment in 2008 and released their first full - length album , " The Shinee World " , on August 28 , 2008 , which won the Newcomer Album of the Year at the 23rd Golden Disk Awards . The first single released from the album was " Sanso Gateun Neo (Love Like Oxygen) " and won first place on " M Countdown " on September 18 , 2008 making it the group 's first win on Korean music shows since debut . Their second album " Lucifer " (2010) produced two singles , " Lucifer " and " Hello " . For their outstanding choreography the group was nominated for the Best Dance Performance Award at the Mnet Asian Music Awards in 2010 . " Lucifer " also won the Disk Bonsang Award at the 25th Golden Disk Awards as well as the Popularity Award . On March 21 , 2012 the group released their fourth EP " Sherlock " for which the group was awarded another Disk Bonsang Award at the 27th Golden Disc Awards

and the Bonsang Award at the 22nd Seoul Music Award . Also following the success of the lead single it was also nominated for Song of the Year at the 2012 Mnet Asian Music Awards . <t> Cho Kyuhyun </t> Cho Kyu - hyun (born February 3 , 1988) , better known mononymously as Kyuhyun , is a South Korean singer and musical theatre actor . He is best known as a member of South Korean boy group Super Junior , its sub - groups Super Junior - K.R.Y. , Super Junior - M and a former member of the South Korean ballad group S.M. the Ballad . He is one of the first four Korean artists to appear on Chinese postage stamps . <t> 2014 S / S </t> 2014 S / S is the debut album of South Korean group WINNER . It was released on August 12 , 2014 by the group 's record label , YG Entertainment . The members were credited for writing the lyrics and composing the majority of the album 's songs . <t> History (band) </t> History (Korean : 히스토리) was a South Korean boy group formed by LOEN Entertainment in 2013 . They debuted on April 26 , 2013 with " Dreamer " , featuring the narration of their labelmate IU . They were LOEN Entertainment 's first boy group . They officially disbanded on May 12 , 2017 . <t> Winner (band) </t> Winner (Hangul : 위너) , often stylized as WINNER , is a South Korean boy group formed in 2013 by YG Entertainment and debuted in 2014 . It currently consists of four members , Jinwoo , Seunghoon , Mino and Seungyoon . Originally a five - piece group with Taehyun , who later departed from the group in November 2016 . <t> Madtown </t> Madtown (Hangul : 매드타운) , often stylized as MADTOWN , is a South Korean boy group formed in 2014 by J. Tune Camp . The group consists of Moos , Daewon , Lee Geon , Jota , Heo Jun , Buffy and H.O. Their debut album , " Mad Town " , was released on October 6 , 2014 . Two of the members , Moos and Buffy , originally debuted as the hip hop duo " Pro C " in 2013 . Madtown 's official fan - base name is Mad - people . Starting December 22 , 2016 , MADTOWN 's contract was sold to GNI Entertainment after J. Tune Camp closed . <t> List of songs written by Ravi </t> Ravi is a South Korean rapper , songwriter and producer , signed under Jellyfish Entertainment . He began his career as a rapper in 2012 in the South Korean boy group VIXX , and later formed VIXX 's first sub - unit VIXX LR with band mate Leo in 2015 . Ravi 's songwriting career began with his participation in co - writing VIXX 's debut single " Super Hero " . As of November 2016 with the release of " VIXX 2016 Conception Ker " , Ravi has contributed to the writing and composing of over 46 songs recorded by VIXX . Ravi

is widely known for his participation of composing and songwriting rap portions for the group as well as lyrics and music . <t> SF9 (band) </t> SF9 (Korean : 에스에프나인 ; shortened from Sensational Feeling 9) is a South Korean boy group formed by FNC Entertainment . SF9 is the company 's first dance boy group to ever debut . SF9 debuted on October 5 , 2016 with the release of their first single album " Feeling Sensation " . <t> Seventeen discography </t> This is the discography of South Korean boy group Seventeen . Seventeen (Hangul : 세븐틴) , also stylized as SEVENTEEN or SVT , is a South Korean boy group formed by Pledis Entertainment in 2015 . They have released one album and four EPs . <t> BTS discography </t> The following is the discography of South Korean boy group BTS . The group debuted in South Korea on June 2013 with single album , " 2 Cool 4 Skool " , at number 5 on South Korean Week 31 Gaon Weekly Chart . They made a comeback on September 2013 with an extended play , " O!RUL8,2 ? " , which peaked at number 4 on Week 38 Gaon Weekly Chart . BTS then released their second extended play , " Skool Luv Affair " , in February 2014 , where it charted at number 1 on Week 18 Gaon Weekly Chart . This also marked the first time their album charted on international charts , Billboard World Albums and Japan 's Oricon Chart , specifically . A repackaged version of the album , " Skool Luv Affair Special Addition " which was released in May 2014 , also peaked at number 1 on Week 21 Gaon Weekly Chart .

B.2 Maps for Question 5a85ea095542994775f606a8

Cell #1

Question: What science fantasy young adult series , told in first person , has a set of companion books narrating the stories of enslaved worlds and alien species ?

Documents: <t> Andre Norton Award </t> The Andre Norton Award for Young Adult Science Fiction and Fantasy is an annual award presented by the Science Fiction and Fantasy Writers of America (SFWA) to the author of the best young adult or middle grade science fiction or fantasy book published in the United States in the preceding year . It is named to honor prolific science fiction and fantasy author Andre Norton (1912–2005) , and it was established by then

SFWA president Catherine Asaro and the SFWA Young Adult Fiction committee and announced on February 20 , 2005 . Any published young adult or middle grade science fiction or fantasy novel is eligible for the prize , including graphic novels . There is no limit on word count . The award is presented along with the Nebula Awards and follows the same rules for nominations and voting ; as the awards are separate , works may be simultaneously nominated for both the Andre Norton award and a Nebula Award . <t> Victoria Hanley </t> Victoria Hanley is an American young adult fantasy novelist . Her first three books , " The Seer And The Sword " , " The Healer 's Keep " and " The Light Of The Oracle " are companion books to one another . Her newest book (released March 2012) is the sequel of a series , called " Indigo Magic " , published by Egmont USA . She 's also published two non - fiction books through Cotton Wood Press ; called " Seize the Story : A Handbook For Teens Who Like To Write " , and " Wild Ink : A Grownups Guide To Writing Fiction For Teens " . <t> The Hork - Bajir Chronicles </t> The Hork - Bajir Chronicles is the second companion book to the " Animorphs " series , written by K. A. Applegate . With respect to continuity within the series , it takes place before book # 23 , " The Pretender " , although the events told in the story occur between the time of " The Ellimist Chronicles " and " The Andalite Chronicles " . The book is introduced by Tobias , who flies to the valley of the free Hork - Bajir , where Jara Hamee tells him the story of how the Yeerks enslaved the Hork - Bajir , and how Aldrea , an Andalite , and her companion , Dak Hamee , a Hork - Bajir , tried to save their world from the invasion . Jara Hamee 's story is narrated from the points of view of Aldrea , Dak Hamee , and Esplin 9466 , alternating in similar fashion to the " Megamorphs " books . <t> Shadowshaper </t> Shadowshaper is a 2015 American urban fantasy young adult novel written by Daniel José Older . It follows Sierra Santiago , an Afro - Boricua teenager living in Brooklyn . She is the granddaughter of a " shadowshaper " , or a person who infuses art with ancestral spirits . As forces of gentrification invade their community and a mysterious being who appropriates their magic begins to hunt the aging shadowshapers , Sierra must learn about her artistic and spiritual heritage to foil the killer . <t> Left Behind : The Kids </t> " Left Behind : The Kids (stylized as LEFT BEHIND > THE KIDS <) " is a series written by Jerry B. Jenkins , Tim LaHaye , and Chris Fabry . The series consists of 40 short novels aimed primarily at the young adult market based on the adult series Left Behind also written by Jerry B. Jenkins . It follows

a core group of teenagers as they experience the rapture and tribulation , based on scriptures found in the Bible , and background plots introduced in the adult novels . Like the adult series , the books were published by Tyndale House Publishing , and released over the 7 year period of 1997 - 2004 . The series has sold over 11 million copies worldwide . <t> List of Square Enix companion books </t> Dozens of Square Enix companion books have been produced since 1998 , when video game developer Square began to produce books that focused on artwork , developer interviews , and background information on the fictional worlds and characters in its games rather than on gameplay details . The first series of these books was the " Perfect Works " series , written and published by Square subsidiary DigiCube . They produced three books between 1998 and 1999 before the line was stopped in favor of the " Ultimania " (アルティマニア , Arutimania) series , a portmanteau of ultimate and mania . This series of books is written by Studio BentStuff , which had previously written game guides for Square for " Final Fantasy VII " . They were published by DigiCube until the company was dissolved in 2003 . Square merged with video game publisher Enix on April 1 , 2003 to form Square Enix , which resumed publication of the companion books . <t> The Divide trilogy </t> The Divide trilogy is a fantasy young adult novel trilogy by Elizabeth Kay , which takes place in an alternate universe . The three books are " The Divide " (2002) , " Back to The Divide " (2005) , and " Jinx on The Divide " (2006) . The first novel was originally published by the small press publisher Chicken House (now a division of Scholastic) , with subsequent volumes published by Scholastic , which also reprinted the first novel . The books have been translated into French , German , Spanish , Finnish , Chinese , Japanese , Portuguese , Italian , Romanian and Dutch . Interior illustrations are by Ted Dewan . <t> Science Fantasy (magazine) </t> Science Fantasy , which also appeared under the titles Impulse and SF Impulse , was a British fantasy and science fiction magazine , launched in 1950 by Nova Publications as a companion to Nova 's " New Worlds " . Walter Gillings was editor for the first two issues , and was then replaced by John Carnell , the editor of " New Worlds " , as a cost - saving measure . Carnell edited both magazines until Nova went out of business in early 1964 . The titles were acquired by Roberts & Vinter , who hired Kyril Bonfiglioli to edit " Science Fantasy " ; Bonfiglioli changed the title to " Impulse " in early 1966 , but the new title led to confusion with the distributors and sales fell , though the magazine remained profitable .

The title was changed again to " SF Impulse " for the last few issues . " Science Fantasy " ceased publication the following year , when Roberts & Vinter came under financial pressure after their printer went bankrupt . <t> Animorphs </t> Animorphs is a science fantasy series of young adult books written by Katherine Applegate and her husband Michael Grant , writing together under the name K. A. Applegate , and published by Scholastic . It is told in first person , with all six main characters taking turns narrating the books through their own perspectives . Horror , war , dehumanization , sanity , morality , innocence , leadership , freedom and growing up are the core themes of the series . <t> Etiquette & amp ; Espionage </t> Etiquette & Espionage is a young adult steampunk novel by Gail Carriger . It is her first young adult novel , and is set in the same universe as her bestselling Parasol Protectorate adult series .

Cell #2

Question: What science fantasy young adult series , told in first person , has a set of companion books narrating the stories of enslaved worlds and alien species ?

Documents: <t> Andre Norton Award </t> The Andre Norton Award for Young Adult Science Fiction and Fantasy is an annual award presented by the Science Fiction and Fantasy Writers of America (SFWA) to the author of the best young adult or middle grade science fiction or fantasy book published in the United States in the preceding year . It is named to honor prolific science fiction and fantasy author Andre Norton (1912–2005) , and it was established by then SFWA president Catherine Asaro and the SFWA Young Adult Fiction committee and announced on February 20 , 2005 . Any published young adult or middle grade science fiction or fantasy novel is eligible for the prize , including graphic novels . There is no limit on word count . The award is presented along with the Nebula Awards and follows the same rules for nominations and voting ; as the awards are separate , works may be simultaneously nominated for both the Andre Norton award and a Nebula Award . <t> Victoria Hanley </t> Victoria Hanley is an American young adult fantasy novelist . Her first three books , " The Seer And The Sword " , " The Healer 's Keep " and " The Light Of The Oracle " are companion books to one another . Her newest book (released March 2012) is the sequel of a series , called " Indigo Magic " , published by Egmont USA . She 's also published two non - fiction books through Cotton Wood Press ; called

" Seize the Story : A Handbook For Teens Who Like To Write " , and " Wild Ink : A Grownups Guide To Writing Fiction For Teens " . <t> The Hork - Bajir Chronicles </t> The Hork - Bajir Chronicles is the second companion book to the " Animorphs " series , written by K. A. Applegate . With respect to continuity within the series , it takes place before book # 23 , " The Pretender " , although the events told in the story occur between the time of " The Ellimist Chronicles " and " The Andalite Chronicles " . The book is introduced by Tobias , who flies to the valley of the free Hork - Bajir , where Jara Hamee tells him the story of how the Yeerks enslaved the Hork - Bajir , and how Aldrea , an Andalite , and her companion , Dak Hamee , a Hork - Bajir , tried to save their world from the invasion . Jara Hamee 's story is narrated from the points of view of Aldrea , Dak Hamee , and Esplin 9466 , alternating in similar fashion to the " Megamorphs " books . <t> Shadowshaper </t> Shadowshaper is a 2015 American urban fantasy young adult novel written by Daniel José Older . It follows Sierra Santiago , an Afro - Boricua teenager living in Brooklyn . She is the granddaughter of a " shadowshaper " , or a person who infuses art with ancestral spirits . As forces of gentrification invade their community and a mysterious being who appropriates their magic begins to hunt the aging shadowshapers , Sierra must learn about her artistic and spiritual heritage to foil the killer . <t> Left Behind : The Kids </t> " Left Behind : The Kids (stylized as LEFT BEHIND > THE KIDS <) " is a series written by Jerry B. Jenkins , Tim LaHaye , and Chris Fabry . The series consists of 40 short novels aimed primarily at the young adult market based on the adult series Left Behind also written by Jerry B. Jenkins . It follows a core group of teenagers as they experience the rapture and tribulation , based on scriptures found in the Bible , and background plots introduced in the adult novels . Like the adult series , the books were published by Tyndale House Publishing , and released over the 7 year period of 1997 - 2004 . The series has sold over 11 million copies worldwide . <t> List of Square Enix companion books </t> Dozens of Square Enix companion books have been produced since 1998 , when video game developer Square began to produce books that focused on artwork , developer interviews , and background information on the fictional worlds and characters in its games rather than on gameplay details . The first series of these books was the " Perfect Works " series , written and published by Square subsidiary DigiCube . They produced three books between 1998 and 1999 before the line was stopped in favor of the " Ultimania " (アルティマニア , Arutimania)

series , a portmanteau of ultimate and mania . This series of books is written by Studio BentStuff , which had previously written game guides for Square for " Final Fantasy VII " . They were published by DigiCube until the company was dissolved in 2003 . Square merged with video game publisher Enix on April 1 , 2003 to form Square Enix , which resumed publication of the companion books . <t> The Divide trilogy </t> The Divide trilogy is a fantasy young adult novel trilogy by Elizabeth Kay , which takes place in an alternate universe . The three books are " The Divide " (2002) , " Back to The Divide " (2005) , and " Jinx on The Divide " (2006) . The first novel was originally published by the small press publisher Chicken House (now a division of Scholastic) , with subsequent volumes published by Scholastic , which also reprinted the first novel . The books have been translated into French , German , Spanish , Finnish , Chinese , Japanese , Portuguese , Italian , Romanian and Dutch . Interior illustrations are by Ted Dewan . <t> Science Fantasy (magazine) </t> Science Fantasy , which also appeared under the titles Impulse and SF Impulse , was a British fantasy and science fiction magazine , launched in 1950 by Nova Publications as a companion to Nova 's " New Worlds " . Walter Gillings was editor for the first two issues , and was then replaced by John Carnell , the editor of " New Worlds " , as a cost - saving measure . Carnell edited both magazines until Nova went out of business in early 1964 . The titles were acquired by Roberts & Vinter , who hired Kyril Bonfiglioli to edit " Science Fantasy " ; Bonfiglioli changed the title to " Impulse " in early 1966 , but the new title led to confusion with the distributors and sales fell , though the magazine remained profitable . The title was changed again to " SF Impulse " for the last few issues . " Science Fantasy " ceased publication the following year , when Roberts & Vinter came under financial pressure after their printer went bankrupt . <t> Animorphs </t> Animorphs is a science fantasy series of young adult books written by Katherine Applegate and her husband Michael Grant , writing together under the name K. A. Applegate , and published by Scholastic . It is told in first person , with all six main characters taking turns narrating the books through their own perspectives . Horror , war , dehumanization , sanity , morality , innocence , leadership , freedom and growing up are the core themes of the series . <t> Etiquette & amp ; Espionage </t> Etiquette & Espionage is a young adult steampunk novel by Gail Carriger . It is her first young adult novel , and is set in the same universe as her bestselling Parasol Protectorate adult series .

B.3 Maps for Question 5a8c7595554299585d9e36b6

Cell #1

Question: What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell ?

Documents: <t> Meet Corliss Archer </t> Meet Corliss Archer , a program from radio 's Golden Age , ran from January 7 , 1943 to September 30 , 1956 . Although it was CBS 's answer to NBC 's popular " A Date with Judy " , it was also broadcast by NBC in 1948 as a summer replacement for " The Bob Hope Show " . From October 3 , 1952 to June 26 , 1953 , it aired on ABC , finally returning to CBS . Despite the program 's long run , fewer than 24 episodes are known to exist . <t> Shirley Temple </t> Shirley Temple Black (April 23 , 1928 – February 10 , 2014) was an American actress , singer , dancer , businesswoman , and diplomat who was Hollywood 's number one box - office draw as a child actress from 1935 to 1938 . As an adult , she was named United States ambassador to Ghana and to Czechoslovakia and also served as Chief of Protocol of the United States . <t> Janet Waldo </t> Janet Marie Waldo (February 4 , 1920 – June 12 , 2016) was an American radio and voice actress . She is best known in animation for voicing Judy Jetson , Nancy in " Shazzan " , Penelope Pitstop , and Josie in " Josie and the Pussycats " , and on radio as the title character in " Meet Corliss Archer " . <t> Meet Corliss Archer (TV series) </t> Meet Corliss Archer is an American television sitcom that aired on CBS (July 13 , 1951 - August 10 , 1951) and in syndication via the Ziv Company from April to December 1954 . The program was an adaptation of the radio series of the same name , which was based on a series of short stories by F. Hugh Herbert . <t> Lord High Treasurer </t> The post of Lord High Treasurer or Lord Treasurer was an English government position and has been a British government position since the Acts of Union of 1707 . A holder of the post would be the third - highest - ranked Great Officer of State , below the Lord High Steward and the Lord High Chancellor . <t> A Kiss for Corliss </t> A Kiss for Corliss is a 1949 American comedy film directed by Richard Wallace and written by Howard Dimsdale . It stars Shirley Temple in her final starring role as well as her final film appearance . It is a sequel to the 1945 film " Kiss and Tell " . " A Kiss for Corliss " was retitled " Almost a Bride " before release and this title appears in the title sequence . The film was released on November 25 , 1949 , by United Artists . <t> Kiss

and Tell (1945 film) </t> Kiss and Tell is a 1945 American comedy film starring then 17-year - old Shirley Temple as Corliss Archer . In the film , two teenage girls cause their respective parents much concern when they start to become interested in boys . The parents ' bickering about which girl is the worse influence causes more problems than it solves . <t> Secretary of State for Constitutional Affairs </t> The office of Secretary of State for Constitutional Affairs was a British Government position , created in 2003 . Certain functions of the Lord Chancellor which related to the Lord Chancellor 's Department were transferred to the Secretary of State . At a later date further functions were also transferred to the Secretary of State for Constitutional Affairs from the First Secretary of State , a position within the government held by the Deputy Prime Minister . <t> Village accountant </t> The Village Accountant (variously known as " Patwari " , " Talati " , " Patel " , " Karnam " , " Adhikari " , " Shanbogaru","Patnaik " etc .) is an administrative government position found in rural parts of the Indian sub - continent . The office and the officeholder are called the " patwari " in Telangana , Bengal , North India and in Pakistan while in Sindh it is called " tapedar " . The position is known as the " karnam " in Andhra Pradesh , " patnaik " in Orissa or " adhikari " in Tamil Nadu , while it is commonly known as the " talati " in Karnataka , Gujarat and Maharashtra . The position was known as the " kulkarni " in Northern Karnataka and Maharashtra . The position was known as the " shanbogaru " in South Karnataka . <t> Charles Craft </t> Charles Craft (May 9 , 1902 – September 19 , 1968) was an English - born American film and television editor . Born in the county of Hampshire in England on May 9 , 1902 , Craft would enter the film industry in Hollywood in 1927 . The first film he edited was the Universal Pictures silent film , " Painting the Town " . Over the next 25 years , Craft would edit 90 feature - length films . In the early 1950s he would switch his focus to the small screen , his first show being " Racket Squad " , from 1951–53 , for which he was the main editor , editing 93 of the 98 episodes . He would work on several other series during the 1950s , including " Meet Corliss Archer " (1954) , " Science Fiction Theatre " (1955–56) , and " Highway Patrol " (1955–57) . In the late 1950s and early 1960s he was one of the main editors on " Sea Hunt " , starring Lloyd Bridges , editing over half of the episodes . His final film work would be editing " Flipper 's New Adventure " (1964 , the sequel to 1963 's " Flipper " . When the film was made into a television series , Craft would begin the editing duties on that show , editing the

first 28 episodes before he retired in 1966 . Craft died on September 19 , 1968 in Los Angeles , California .

Cell #2

Question: What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell ?

Documents: <t> Meet Corliss Archer </t> Meet Corliss Archer , a program from radio 's Golden Age , ran from January 7 , 1943 to September 30 , 1956 . Although it was CBS 's answer to NBC 's popular " A Date with Judy " , it was also broadcast by NBC in 1948 as a summer replacement for " The Bob Hope Show " . From October 3 , 1952 to June 26 , 1953 , it aired on ABC , finally returning to CBS . Despite the program 's long run , fewer than 24 episodes are known to exist . <t> Shirley Temple </t> Shirley Temple Black (April 23 , 1928 – February 10 , 2014) was an American actress , singer , dancer , businesswoman , and diplomat who was Hollywood 's number one box - office draw as a child actress from 1935 to 1938 . As an adult , she was named United States ambassador to Ghana and to Czechoslovakia and also served as Chief of Protocol of the United States . <t> Janet Waldo </t> Janet Marie Waldo (February 4 , 1920 – June 12 , 2016) was an American radio and voice actress . She is best known in animation for voicing Judy Jetson , Nancy in " Shazzan " , Penelope Pitstop , and Josie in " Josie and the Pussycats " , and on radio as the title character in " Meet Corliss Archer " . <t> Meet Corliss Archer (TV series) </t> Meet Corliss Archer is an American television sitcom that aired on CBS (July 13 , 1951 - August 10 , 1951) and in syndication via the Ziv Company from April to December 1954 . The program was an adaptation of the radio series of the same name , which was based on a series of short stories by F. Hugh Herbert . <t> Lord High Treasurer </t> The post of Lord High Treasurer or Lord Treasurer was an English government position and has been a British government position since the Acts of Union of 1707 . A holder of the post would be the third - highest - ranked Great Officer of State , below the Lord High Steward and the Lord High Chancellor . <t> A Kiss for Corliss </t> A Kiss for Corliss is a 1949 American comedy film directed by Richard Wallace and written by Howard Dimsdale . It stars Shirley Temple in her final starring role as well as her final film appearance . It is a sequel to the 1945 film " Kiss and

Tell " . " A Kiss for Corliss " was retitled " Almost a Bride " before release and this title appears in the title sequence . The film was released on November 25 , 1949 , by United Artists . <t> Kiss and Tell (1945 film) </t> Kiss and Tell is a 1945 American comedy film starring then 17-year - old Shirley Temple as Corliss Archer . In the film , two teenage girls cause their respective parents much concern when they start to become interested in boys . The parents ' bickering about which girl is the worse influence causes more problems than it solves . <t> Secretary of State for Constitutional Affairs </t> The office of Secretary of State for Constitutional Affairs was a British Government position , created in 2003 . Certain functions of the Lord Chancellor which related to the Lord Chancellor 's Department were transferred to the Secretary of State . At a later date further functions were also transferred to the Secretary of State for Constitutional Affairs from the First Secretary of State , a position within the government held by the Deputy Prime Minister . <t> Village accountant </t> The Village Accountant (variously known as " Patwari " , " Talati " , " Patel " , " Karnam " , " Adhikari " , " Shanbogaru","Patnaik " etc .) is an administrative government position found in rural parts of the Indian sub - continent . The office and the officeholder are called the " patwari " in Telangana , Bengal , North India and in Pakistan while in Sindh it is called " tapedar " . The position is known as the " karnam " in Andhra Pradesh , " patnaik " in Orissa or " adhikari " in Tamil Nadu , while it is commonly known as the " talati " in Karnataka , Gujarat and Maharashtra . The position was known as the " kulkarni " in Northern Karnataka and Maharashtra . The position was known as the " shanbogaru " in South Karnataka . <t> Charles Craft </t> Charles Craft (May 9 , 1902 – September 19 , 1968) was an English - born American film and television editor . Born in the county of Hampshire in England on May 9 , 1902 , Craft would enter the film industry in Hollywood in 1927 . The first film he edited was the Universal Pictures silent film , " Painting the Town " . Over the next 25 years , Craft would edit 90 feature - length films . In the early 1950s he would switch his focus to the small screen , his first show being " Racket Squad " , from 1951–53 , for which he was the main editor , editing 93 of the 98 episodes . He would work on several other series during the 1950s , including " Meet Corliss Archer " (1954) , " Science Fiction Theatre " (1955–56) , and " Highway Patrol " (1955–57) . In the late 1950s and early 1960s he was one of the main editors on " Sea Hunt " , starring Lloyd Bridges , editing over half of the episodes . His final film work would be editing

" Flipper 's New Adventure " (1964 , the sequel to 1963 's " Flipper " . When the film was made into a television series , Craft would begin the editing duties on that show , editing the first 28 episodes before he retired in 1966 . Craft died on September 19 , 1968 in Los Angeles , California .

B.4 Maps for Question 5a8b57f25542995d1e6f1371

Cell #1

Question: Were Scott Derrickson and Ed Wood of the same nationality ?

Documents: <t> Ed Wood (film) </t> Ed Wood is a 1994 American biographical period comedy - drama film directed and produced by Tim Burton , and starring Johnny Depp as cult filmmaker Ed Wood . The film concerns the period in Wood 's life when he made his best - known films as well as his relationship with actor Bela Lugosi , played by Martin Landau . Sarah Jessica Parker , Patricia Arquette , Jeffrey Jones , Lisa Marie , and Bill Murray are among the supporting cast . <t> Scott Derrickson </t> Scott Derrickson (born July 16 , 1966) is an American director , screenwriter and producer . He lives in Los Angeles , California . He is best known for directing horror films such as " Sinister " , " The Exorcism of Emily Rose " , and " Deliver Us From Evil " , as well as the 2016 Marvel Cinematic Universe installment , " Doctor Strange . " <t> Woodson , Arkansas </t> Woodson is a census - designated place (CDP) in Pulaski County , Arkansas , in the United States . Its population was 403 at the 2010 census . It is part of the Little Rock – North Little Rock – Conway Metropolitan Statistical Area . Woodson and its accompanying Woodson Lake and Wood Hollow are the namesake for Ed Wood Sr . , a prominent plantation owner , trader , and businessman at the turn of the 20th century . Woodson is adjacent to the Wood Plantation , the largest of the plantations own by Ed Wood Sr . <t> Tyler Bates </t> Tyler Bates (born June 5 , 1965) is an American musician , music producer , and composer for films , television , and video games . Much of his work is in the action and horror film genres , with films like " Dawn of the Dead , 300 , Sucker Punch , " and " John Wick . " He has collaborated with directors like Zack Snyder , Rob Zombie , Neil Marshall , William Friedkin , Scott Derrickson

, and James Gunn . With Gunn , he has scored every one of the director 's films ; including " Guardians of the Galaxy " , which became one of the highest grossing domestic movies of 2014 , and its 2017 sequel . In addition , he is also the lead guitarist of the American rock band Marilyn Manson , and produced its albums " The Pale Emperor " and " Heaven Upside Down " . <t> Ed Wood </t> Edward Davis Wood Jr. (October 10 , 1924 – December 10 , 1978) was an American filmmaker , actor , writer , producer , and director . <t> Deliver Us from Evil (2014 film) </t> Deliver Us from Evil is a 2014 American supernatural horror film directed by Scott Derrickson and produced by Jerry Bruckheimer . The film is officially based on a 2001 non - fiction book entitled " Beware the Night " by Ralph Sarchie and Lisa Collier Cool , and its marketing campaign highlighted that it was " inspired by actual accounts " . The film stars Eric Bana , Édgar Ramírez , Sean Harris , Olivia Munn , and Joel McHale in the main roles and was released on July 2 , 2014 . <t> Adam Collis </t> Adam Collis is an American filmmaker and actor . He attended the Duke University from 1986 to 1990 and the University of California , Los Angeles from 2007 to 2010 . He also studied cinema at the University of Southern California from 1991 to 1997 . Collis first work was the assistant director for the Scott Derrickson 's short " Love in the Ruins " (1995) . In 1998 , he played " Crankshaft " in Eric Koyanagi 's " Hundred Percent " . <t> Sinister (film) </t> Sinister is a 2012 supernatural horror film directed by Scott Derrickson and written by Derrickson and C. Robert Cargill . It stars Ethan Hawke as fictional true - crime writer Ellison Oswalt who discovers a box of home movies in his attic that puts his family in danger . <t> Conrad Brooks </t> Conrad Brooks (born Conrad Biedrzycki on January 3 , 1931 in Baltimore , Maryland) is an American actor . He moved to Hollywood , California in 1948 to pursue a career in acting . He got his start in movies appearing in Ed Wood films such as " Plan 9 from Outer Space " , " Glen or Glenda " , and " Jail Bait . " He took a break from acting during the 1960s and 1970s but due to the ongoing interest in the films of Ed Wood , he reemerged in the 1980s and has become a prolific actor . He also has since gone on to write , produce and direct several films . <t> Doctor Strange (2016 film) </t> Doctor Strange is a 2016 American superhero film based on the Marvel Comics character of the same name , produced by Marvel Studios and distributed by Walt Disney Studios Motion Pictures . It is the fourteenth film of the Marvel Cinematic Universe (MCU) . The film was directed by Scott Derrickson ,

who wrote it with Jon Spaihts and C. Robert Cargill , and stars Benedict Cumberbatch as Stephen Strange , along with Chiwetel Ejiofor , Rachel McAdams , Benedict Wong , Michael Stuhlbarg , Benjamin Bratt , Scott Adkins , Mads Mikkelsen , and Tilda Swinton . In " Doctor Strange " , surgeon Strange learns the mystic arts after a career - ending car accident .

Cell #2

Question: Were Scott Derrickson and Ed Wood of the same nationality ?

Documents: <t> Ed Wood (film) </t> Ed Wood is a 1994 American biographical period comedy - drama film directed and produced by Tim Burton , and starring Johnny Depp as cult filmmaker Ed Wood . The film concerns the period in Wood 's life when he made his best - known films as well as his relationship with actor Bela Lugosi , played by Martin Landau . Sarah Jessica Parker , Patricia Arquette , Jeffrey Jones , Lisa Marie , and Bill Murray are among the supporting cast . <t> Scott Derrickson </t> Scott Derrickson (born July 16 , 1966) is an American director , screenwriter and producer . He lives in Los Angeles , California . He is best known for directing horror films such as " Sinister " , " The Exorcism of Emily Rose " , and " Deliver Us From Evil " , as well as the 2016 Marvel Cinematic Universe installment , " Doctor Strange . " <t> Woodson , Arkansas </t> Woodson is a census - designated place (CDP) in Pulaski County , Arkansas , in the United States . Its population was 403 at the 2010 census . It is part of the Little Rock – North Little Rock – Conway Metropolitan Statistical Area . Woodson and its accompanying Woodson Lake and Wood Hollow are the namesake for Ed Wood Sr . , a prominent plantation owner , trader , and businessman at the turn of the 20th century . Woodson is adjacent to the Wood Plantation , the largest of the plantations own by Ed Wood Sr . <t> Tyler Bates </t> Tyler Bates (born June 5 , 1965) is an American musician , music producer , and composer for films , television , and video games . Much of his work is in the action and horror film genres , with films like " Dawn of the Dead , 300 , Sucker Punch , " and " John Wick . " He has collaborated with directors like Zack Snyder , Rob Zombie , Neil Marshall , William Friedkin , Scott Derrickson , and James Gunn . With Gunn , he has scored every one of the director 's films ; including " Guardians of the Galaxy " , which became one of the highest grossing domestic movies of 2014 , and its 2017 sequel . In addition , he is also the lead guitarist of the American rock band

Marilyn Manson , and produced its albums " The Pale Emperor " and " Heaven Upside Down " . <t> Ed Wood </t> Edward Davis Wood Jr. (October 10 , 1924 – December 10 , 1978) was an American filmmaker , actor , writer , producer , and director . <t> Deliver Us from Evil (2014 film) </t> Deliver Us from Evil is a 2014 American supernatural horror film directed by Scott Derrickson and produced by Jerry Bruckheimer . The film is officially based on a 2001 non - fiction book entitled " Beware the Night " by Ralph Sarchie and Lisa Collier Cool , and its marketing campaign highlighted that it was " inspired by actual accounts " . The film stars Eric Bana , Édgar Ramírez , Sean Harris , Olivia Munn , and Joel McHale in the main roles and was released on July 2 , 2014 . <t> Adam Collis </t> Adam Collis is an American filmmaker and actor . He attended the Duke University from 1986 to 1990 and the University of California , Los Angeles from 2007 to 2010 . He also studied cinema at the University of Southern California from 1991 to 1997 . Collis first work was the assistant director for the Scott Derrickson 's short " Love in the Ruins " (1995) . In 1998 , he played " Crankshaft " in Eric Koyanagi 's " Hundred Percent " . <t> Sinister (film) </t> Sinister is a 2012 supernatural horror film directed by Scott Derrickson and written by Derrickson and C. Robert Cargill . It stars Ethan Hawke as fictional true - crime writer Ellison Oswalt who discovers a box of home movies in his attic that puts his family in danger . <t> Conrad Brooks </t> Conrad Brooks (born Conrad Biedrzycki on January 3 , 1931 in Baltimore , Maryland) is an American actor . He moved to Hollywood , California in 1948 to pursue a career in acting . He got his start in movies appearing in Ed Wood films such as " Plan 9 from Outer Space " , " Glen or Glenda " , and " Jail Bait . " He took a break from acting during the 1960s and 1970s but due to the ongoing interest in the films of Ed Wood , he reemerged in the 1980s and has become a prolific actor . He also has since gone on to write , produce and direct several films . <t> Doctor Strange (2016 film) </t> Doctor Strange is a 2016 American superhero film based on the Marvel Comics character of the same name , produced by Marvel Studios and distributed by Walt Disney Studios Motion Pictures . It is the fourteenth film of the Marvel Cinematic Universe (MCU) . The film was directed by Scott Derrickson , who wrote it with Jon Spaihts and C. Robert Cargill , and stars Benedict Cumberbatch as Stephen Strange , along with Chiwetel Ejiofor , Rachel McAdams , Benedict Wong , Michael Stuhlbarg

, Benjamin Bratt , Scott Adkins , Mads Mikkelsen , and Tilda Swinton . In " Doctor Strange " , surgeon Strange learns the mystic arts after a career - ending car accident .

APPENDIX C

BERT-based Model Attention Maps

In this appendix we provide full attention maps of 8 HotpotQA dev set questions from our BERT-based document selection model. Colour indicates attention values, with darker red indicating higher attention values. Questions are referred to by their HotpotQA ID.

C.1 Maps for Question 5a8b57f25542995d1e6f1371

Question ID 5a8b57f25542995d1e6f1371

Cell #1

Control: were scott derrick ##son and ed wood of the same nationality ?

Read: < t > scott derrick ##son < / t > scott derrick ##son (born july 16 , 1966) is an american director , screenwriter and producer . he lives in los angeles , california . he is best known for directing horror films such as " sinister " , " the ex ##or ##cis ##m of emily rose " , and " deliver us from evil " , as well as the 2016 marvel cinematic universe installment , " doctor strange . "

Cell #2

Control: were scott derrick ##son and ed wood of the same nationality ?

Read: < t > ed wood < / t > edward davis wood jr . (october 10 , 1924 – december 10 , 1978) was an american filmmaker , actor , writer , producer , and director .

C.2 Maps for Question 5a8c7595554299585d9e36b6

Question ID 5a8c7595554299585d9e36b6

Cell #1

Control: what government position was held by the woman who portrayed co ##rl ##iss archer in the film kiss and tell ?

Read: < t > kiss and tell (1945 film) < / t > kiss and tell is a 1945 american comedy film starring then 17 - year - old shirley temple as co ##rl ##iss archer . in the film , two teenage girls cause their respective parents much concern when they start to become interested in boys . the parents ' bi ##cker ##ing about which girl is the worse influence causes more problems than it solve ##s .

Cell #2

Control: what government position was held by the woman who portrayed co ##rl ##iss archer in the film kiss and tell ?

Read: < t > shirley temple < / t > shirley temple black (april 23 , 1928 – february 10 , 2014) was an american actress , singer , dancer , business ##woman , and diplomat who was hollywood ' s number one box - office draw as a child actress from 1935 to 1938 . as an adult , she was named united states ambassador to ghana and to czechoslovakia and also served as chief of protocol of the united states .

C.3 Maps for Question 5a85ea095542994775f606a8

Question ID 5a85ea095542994775f606a8

Cell #1

Control: what science fantasy young adult series , told in first person , has a set of companion books na ##rra ##ting the stories of enslaved worlds and alien species ?

Read: < t > the ho ##rk - ba ##ji ##r chronicles < / t > the ho ##rk - ba ##ji ##r chronicles is the second companion book to the " an ##imo ##rp ##hs " series , written by k . a . apple ##gate . with respect to continuity within the series , it takes place before book # 23 , " the pretend ##er " , although the events told in the story occur between the time of " the el ##lim ##ist chronicles " and " the and ##ali ##te chronicles " . the book is introduced by tobias , who flies to the valley of the free ho ##rk - ba ##ji ##r , where jar ##a ham ##ee tells him the story of how the ye ##er ##ks enslaved the ho ##rk - ba ##ji ##r , and how al ##dre ##a , an and ##ali ##te , and her companion , da ##k ham ##ee , a ho ##rk - ba ##ji ##r , tried to save their world from the invasion . jar ##a ham ##ee ' s story is narrated from the points of view of al ##dre ##a , da ##k ham ##ee , and es ##plin 94 ##66 , alternating in similar fashion to the " mega ##mo ##rp ##hs " books .

Cell #2

Control: what science fantasy young adult series , told in first person , has a set of companion books na ##rra ##ting the stories of enslaved worlds and alien species ?

Read: < t > an ##imo ##rp ##hs < / t > an ##imo ##rp ##hs is a science fantasy series of young adult books written by katherine apple ##gate and her husband michael grant , writing together under the name k . a . apple ##gate , and published by scholastic . it is told in first person , with all six main characters taking turns na ##rra ##ting the books through their own perspectives . horror , war , de ##hum ##ani ##zation , sanity , morality , innocence , leadership , freedom and growing up are the core themes of the series .

C.4 Maps for Question 5adbf0a255429947ff17385a

Question ID 5adbf0a255429947ff17385a

Cell #1

Control: are the lal ##eli mosque and es ##ma sultan mansion located in the same neighborhood ?

Read: < t > es ##ma sultan mansion < / t > the es ##ma sultan mansion (turkish : " es ##ma sultan ya ##l ##ı ##s ##ı ") , a historical ya ##l ##ı (english : waters ##ide mansion) located at bo ##sp ##hor ##us in or ##ta ##ko ##y neighborhood of istanbul , turkey and named after its original owner es ##ma sultan , is used today as a cultural center after being redeveloped .

Cell #2

Control: are the lal ##eli mosque and es ##ma sultan mansion located in the same neighborhood ?

Read: < t > lal ##eli mosque < / t > the lal ##eli mosque (turkish : " lal ##eli cam ##ii , or tu ##lip mosque ") is an 18th - century ottoman imperial mosque located in lal ##eli , fat ##ih , istanbul , turkey .

C.5 Maps for Question 5a8e3ea95542995a26add48d

Question ID 5a8e3ea95542995a26add48d

Cell #1

Control: the director of the romantic comedy " big stone gap " is based in what new york city ?

Read: < t > big stone gap (film) < / t > big stone gap is a 2014 american drama romantic comedy film written and directed by adrian ##a tri ##gia ##ni and produced by donna gig ##lio ##tti for altar identity studios , a subsidiary of media society . based on tri ##gia ##ni ' s 2000 best - selling novel of the same name , the story is set in the actual virginia town of big stone gap circa 1970s . the film had its world premiere at the virginia film festival on november 6 , 2014 .

Cell #2

Control: the director of the romantic comedy " big stone gap " is based in what new york city ?

Read: < t > adrian ##a tri ##gia ##ni < / t > adrian ##a tri ##gia ##ni is an italian american best - selling author of sixteen books , television writer , film director , and entrepreneur based in greenwich village , new york city . tri ##gia ##ni has published a novel a year since 2000 .

C.6 Maps for Question 5abd94525542992ac4f382d2

Question ID 5abd94525542992ac4f382d2

Cell #1

Control: 2014 s / s is the debut album of a south korean boy group that was formed by who ?

Read: < t > 2014 s / s < / t > 2014 s / s is the debut album of south korean group winner . it was released on august 12 , 2014 by the group ' s record label , y ##g entertainment . the members were credited for writing the lyrics and composing the majority of the album ' s songs .

Cell #2

Control: 2014 s / s is the debut album of a south korean boy group that was formed by who ?

Read: < t > winner (band) < / t > winner (hangul : [UNK]) , often stylized as winner , is a south korean boy group formed in 2013 by y ##g entertainment and debuted in 2014 . it currently consists of four members , jin ##wo ##o , se ##ung ##ho ##on , min ##o and se ##ung ##yo ##on . originally a five - piece group with tae ##hy ##un , who later departed from the group in november 2016 .

C.7 Maps for Question 5a85b2d95542997b5ce40028

Question ID 5a85b2d95542997b5ce40028

Cell #1

Control: who was known by his stage name ala ##din and helped organizations improve their performance as a consultant ?

Read: < t > management consulting < / t > management consulting is the practice of helping organizations to improve their performance , operating primarily through the analysis of existing organizational problems and the development of plans for improvement . organizations may draw upon the services of management consultants for a number of reasons , including gaining external (and presumably objective) advice and access to the consultants ' specialized expertise .

Cell #2

Control: who was known by his stage name ala ##din and helped organizations improve their performance as a consultant ?

Read: < t > ee ##nas ##ul fate ##h < / t > ee ##nas ##ul fate ##h (bengali : [UNK] [UNK] ; born 3 april 1959) , also known by his stage name ala ##din , is a bangladeshi - british cultural practitioner , magician , live artist and former international management consultant .

C.8 Maps for Question 5a87ab905542996e4f3088c1

Question ID 5a87ab905542996e4f3088c1

Cell #1

Control: the arena where the lewis ##ton maine ##iac ##s played their home games can seat how many people ?

Read: < t > lewis ##ton maine ##iac ##s < / t > the lewis ##ton maine ##iac ##s were a junior ice hockey team of the quebec major junior hockey league based in lewis ##ton , maine . the team played its home games at the and ##ros ##co ##gg ##in bank coli ##see . they were the second q ##m ##jhl team in the united states , and the only one to play a full season . they won the president ' s cup in 2007 .

Cell #2

Control: the arena where the lewis ##ton maine ##iac ##s played their home games can seat how many people ?

Read: < t > and ##ros ##co ##gg ##in bank coli ##see < / t > the and ##ros ##co ##gg ##in bank coli ##see (formerly central maine civic center and lewis ##ton coli ##see) is a 4 , 000 capacity (3 , 67 ##7 seated) multi - purpose arena , in lewis ##ton , maine , that opened in 1958 . in 1965 it was the location of the world heavyweight title fight during which one of the most famous sports photographs of the century was taken of mu ##ham ##med ali standing over sonny list ##on .