Answering Consumer Health Questions on the Web

by

Amir Vakili Tahami

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Management Sciences

Waterloo, Ontario, Canada, 2022

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contribution

Chapters 3 and 4 describe work related to answer prediction that was done in collaboration with Dake Zhang and Prof. Mark D. Smucker. The idea to aggregate document stances was first proposed by Mark D. Smucker and implemented by Dake Zhang. The work in this thesis on answer aggregation is an extension of that work. Reranking based on stance agreement of documents with questions was first proposed by Ronak Pradeep in the 2020 Health Misinformation track. In this work, we use question-answering instead of stance agreement which is a generalization of that approach.

I am the sole author of the rest of this thesis, written under the guidance of Professor Mark D. Smucker.

This thesis includes work presented in the following notebook papers:

Mustafa Abualsaud, Irene XiangYi Chen, Kamyar Ghajar, Linh Nhi Phan Minh, Mark D. Smucker, Amir Vakili Tahami, and Dake Zhang. UWaterlooMDS at the TREC 2021 Health Misinformation Track. In TREC, 18 pages, 2021.

Amir Vakili Tahami, and Dake Zhang, Mark D. Smucker. UWaterlooMDS at the TREC 2022 Health Misinformation Track. In TREC, 8 pages, 2022.

I understand that my thesis may be made electronically available to the public.

Abstract

Question answering is an important sub task in the field of information retrieval. Question answering has typically used reliable sources of information such as the Wikipedia for information. In this work, we look at answering health questions using the web. The web offers the means to answer general medical questions on a variety of topics, but comes with the downside of being rife with misinformation and contradictory information. We develop our techniques using the TREC health misinformation tracks that use consumer health question as topics and web crawls as their document collection.

In this work, we implement a document filtering technique based on topic-sensitive PageRank that uses a web graph of the hosts in common crawl. We develop a new passage extraction technique that performs query-based contextualized sentence selection. We test this technique on a multi-span extractive question answering dataset. We also develop an answer aggregation technique that can combine language features and manual features to predict answers to these consumer health questions. We test all of these approaches on the TREC Health Misinformation Track. We show that these techniques in the majority of cases provide an uplift in performance.

Acknowledgements

I would like to acknowledge the help and guidance offered by my supervisor Dr. Mark D. Smucker. His assistance and expertise were indispensable throughout my time at the University of Waterloo. I would also like to thank Dr. Charles L.A. Clarke and Dr. Olga Vechtomova for their reviews and comments regarding this work. I would also like to acknowledge the financial support provided by the Graduate Research Studentship Award and the International Student Award provided by the University of Waterloo and the Government of Canada. I would also like to thank Compute Canada for the computational hardware and support they provided.

Dedication

To my family.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Search engines are a modern tool used extensively in our day to day lives. Over the years many advances have been made to the techniques employed by search engines and the search engine results page of today differs greatly from when these tools were first introduced. One example is how modern search engines incorporate question answering techniques into their pipelines, giving short answers in conjunction with a traditional list of web documents. For simple factoid questions, where one or a few words can give a precise answer, Google can give accurate answers based on what it finds on the web as seen in figure 1.1, or for more complicated questions it can extract a summary from a web page as seen in 1.2. Question Answering is one of the principal areas of research within information retrieval and natural language processing.

In this work, we look at automated question answering on the web, more specifically answering consumer health questions using web documents as resources. Question answering on web documents offers a unique set of challenges compared to using knowledge bases or trusted sources like Wikipedia. One key challenge stems from the presence of contradictory answers which would not be present when using a trusted source such as an encyclopedia or a knowledge base. We focus on health-related questions which is a genre of questions that tends to exacerbate this issue. For certain health queries, search engines will return relevant information that is biased towards saying potential treatments are helpful regardless of the truth. This results in the promotion of

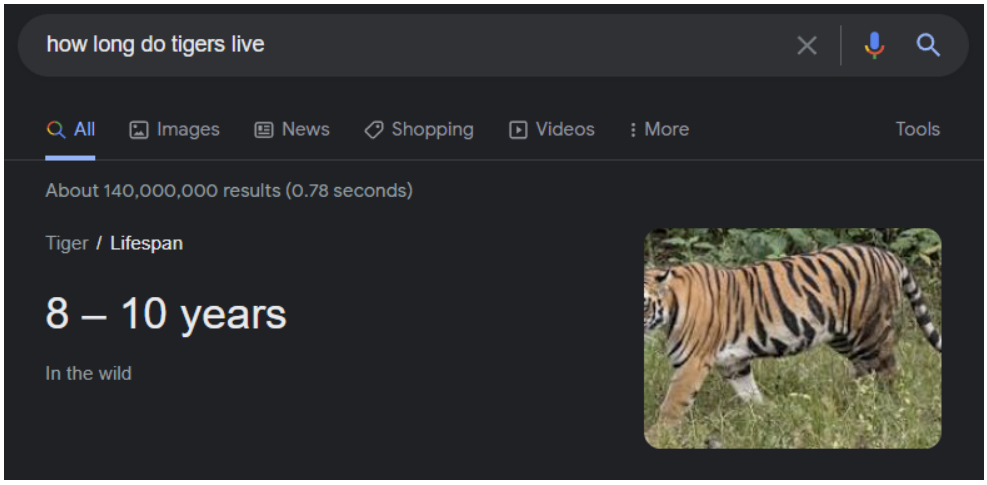

Figure 1.1: Google.com providing a concise answer to a simple factoid question.

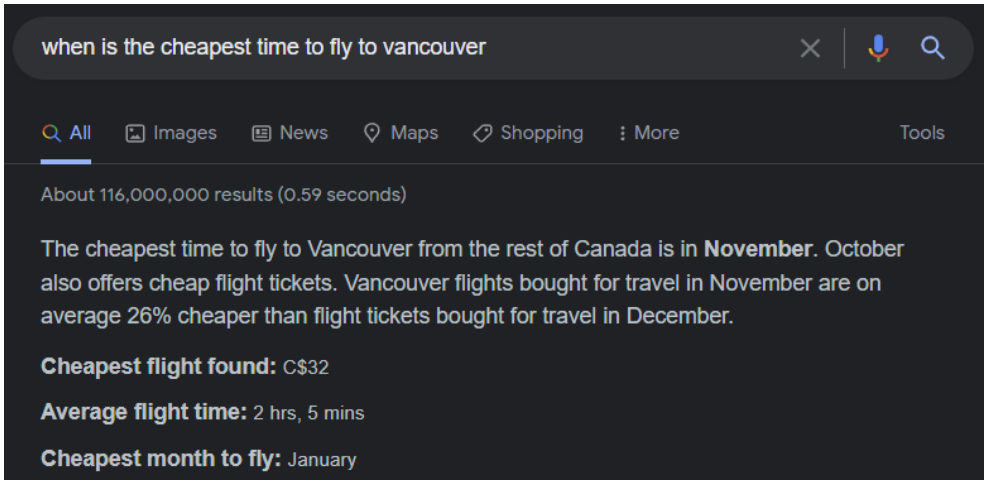

Figure 1.2: Google.com providing an extracted summary of a web page with relevant text highlighted in response to a more complex question.

misinformation or information that is not supported by scientific evidence (White, 2014; White and Hassan, 2014; White and Horvitz, 2015). In such a scenario, where a system is faced with questions where there is the possibility of returning incorrect information, the system would have to first determine the correct answer using a question answering module and promote pages that agree with that answer in its ranking using a reranking module.

1.1 Problem Definition

In Question Answering (QA), the aim is to provide answers to natural language questions. Modern search engines such as Google and Bing combine their search engine results pages with question answering techniques enabling them to answer factoid questions in a concise manner. Factoid questions can be answered precisely using one or a few words.

Question answering systems can either be **extractive** or **generative**. Meaning the answers can be either extracted directly from text or generated in a free-form manner. In this work, we focus on questions with yes or no answers which is a simplification the generative setting.

Open domain question answering (ODQA) is another subset of question answering wherein a system uses external knowledge sources, such as web pages, to automatically answer questions. Open domain question answering differs from the machine reading or machine comprehension task in that no context is provided to derive the answer from. The system must find the relevant information itself within a collection of documents whereas, in machine reading and machine comprehension, a passage is provided to the system for it to extract an answer from. In open domain question answering, the system often has access to an external knowledge source of some kind (open-book setting) although this may not always be the case (closed-book setting).

These systems typically use structured knowledge bases (knowledge graphs) such as DPPedia (Auer et al., 2007) or WikiData (Vrandečić and Krötzsch, 2014). These sources however can be limited and can be difficult to work with for general purpose, non-factoid questions (Zhu et al., 2021).

More advanced ODQA systems use unstructured data sources such as text. For example, they may use the entirety of Wikipedia (Chen et al., 2017), news articles, or science books (Mihaylov et al., 2018). These textual question answering systems are more scalable as such unstructured text sources are more commonly available and cover more subjects (Zhu et al., 2021).

As mentioned, open domain question answering differs from the machine reading comprehension task, which is also a subset of the question answering field, in that no context is provided from which to derive the answer. The ODQA must search for relevant documents in a document collection. Open domain question answering is a more complex task than machine reading. Building a general-purpose accurate open question answering system can be thought of as one of the end goals of the question answering field.

While using unstructured text sources can greatly enhance the capability question answering systems, not all answers can be found in a single source of knowledge such as Wikipedia. In this thesis, we use a large web crawl of approximately 1 billion documents for question answering. Using such a large collection will further increase the variety of questions that a system can answer, but it does come with a significant downside. While sources like Wikipedia can be considered reliable with few mistakes and no disinformation, the same cannot be said for the web as a whole.

So at this point, we reach the main challenge we aim to tackle in this work. How can we correctly determine answers when we may have contradictory information from multiple sources? In the health domain, which will be the primary focus of this work, this problem becomes more severe as incorrect answers can cause actual harm to users. This harm can range from an inconvenience to potentially life-threatening. For example, a search for whether toothpaste will get rid of pimples in commercial search engines will return links to reputable medical websites with experts warning against the practice, but also links to websites demonstrating how to apply technique. In another example, Hoxsey therapy is a hoax treatment marketed as a cure for cancer. A search for "Hoxsey Therapy" in commercial search engines will return links to reputable websites stating it is an ineffective treatment, but also link to websites that sell the formula and talk about it's "anti-tumour" properties.

Another challenge when working on the web is the length of documents. Typically web documents are long and can cover a variety of topics and only a handful of sentences are directly related to answering the question at hand. Most models extract contiguous spans of text or passages from documents for this purpose, but relevant spans can be sparsely distributed across a document. Others have proposed domain specific passage extraction (Zhang et al., 2022). Researchers have shown that individually classifying sentences in isolation is inefficient and gives subpar results and that classifying sentences needs to take the entire context into account (Zhu et al., 2020).

In this work, we implement a fairly simple transformer architecture to classify sentences in a document as relevant or non-relevant to the question at hand. We use a transformer that can take longer inputs so that we can classify each sentence using the context of the entire document. We then use these passages and combine them with document features such as host names or document metadata to get a final prediction of the answer to a question. We implement an answer aggregation technique where we combine transformer outputs of the top k documents. Using this prediction we rerank retrieved documents to display only those that agree with the prediction.

1.2 Thesis Overview

In summary, the challenges tackled and ideas proposed in this work are as follows:

- Reducing harmful information in web retrieval for health questions. Focusing on relevance without paying attention to the correctness of documents could potentially cause harm to users. To tackle the issue of aggregating responses when our knowledge source contains misinformation, we first implemented a simple solution. In order to ensure only reliable sources make it into the final ranking step, we filter domains by their credibility using a web hostname graph and topic-based page rank. For reducing harmful information, previous research indicates that the better approach is to rerank documents based on their level of agreement with the correct answer to the question. In this work, we take previous work done in this regard and expand on

it. This will result in a significant reduction in the amount of harmful information delivered to a user. This approach, however, comes with its new set of challenges.

- When a model needs to determine the answer to a question from a document, it usually does not need all of its text, it only needs certain relevant spans of text. Transformer models used for question answering have input length limits and increased memory requirements for processing long inputs, so shorter passages are preferred to whole documents. For the task of query-biased passage extraction, we implement a model for passage extraction that is both efficient and can extract sentences from throughout the document. The passage extraction is capable of extracting non-continuous spans of text from a document.

- When a user asks a question, we won't have the answer on hand that our system needs for document reranking. Using our passage extraction system we propose a new answer aggregation and reranking architecture for use in the TREC health misinformation tracks. In this approach, we aggregate the outputs of the top 16 retrieved document's transformer models for answer prediction. Combining the passage extraction and this answer aggregation technique we can do reranking based on the agreement between answers derived from document passages and our predicted answer and return those results to the user.

The question answering system in this work will have the following components:

1. A computationally efficient retrieval system that fetches documents relevant to the question from the entire web collection (e.g. BM25 retrieval).

2. An optional computationally expensive reranker that sorts the documents retrieved in the initial stage.

3. An answer aggregation component that determines the answer as yes or no.

4. A final reranker that will rank documents based on their level of agreement with the answer prediction.

We will be using this system to determine the answers to consumer medical questions. Although we constrain the problem to questions with only yes/no answers. The QA system will use documents retrieved from a web crawl. After predicting an answer the system will rerank the retrieved documents to promote the ones that agree with the prediction and suppress those that do not. In this work, we improve or generalize the techniques previously used for each of these modules.

1.3 Contributions

In this section briefly go over the results

- Filtering our collection of web documents to only our seed list of credible domains and using BM25 retrieval improved our main evaluation metric (compatibility) from -0.022 to 0.027. By expanding the list of credible domains using topic-based PageRank we further improved the metric to 0.040. Based on this one might conclude that using more advanced retrieval models on the filtered dataset would improve performance. But we see that there is an upper limit to the quality of results using this technique. Filtering the collection limits the variety of questions we can answer, and for certain topics, we end up with worse performance than had we not filtered the collection. While this approach to the problem does not give better results than answer prediction and reranking strategies, it did yield some insights into the nature of the problem.

- Compared to Mono-T5 (Pradeep et al., 2021b) passage extraction, our passage extraction technique has a delta compatibility of 0.073 vs 0.062 in the 2019 track. In the 2021 track, however, it failed to beat Mono-T5 baseline as its helpfulness compatibility was too low (0.217 vs 0.246).

  We also compare it to the previous best model in the MASH-QA dataset (Zhu et al., 2020), which is a dataset for multi span passage extraction. Using this approach on the MASH-QA dataset we achieve an F1 score of 74.37 vs 57.00 the previous best approach.

- For answer prediction, compared to Zhang et al. (2022) answer prediction, our passage extraction technique gives an AUC score of 0.661 vs 0.606, 0.840 vs 0.822, and 0.691 vs 0.864 in the 2019, 2021, and the 2022 tracks respectively. Compared to Zhang et al. (2022) answer prediction reranking, our passage extraction technique gives a compatibility score of 0.162 vs 0.129, and 0.089 vs 0.076 in the 2021 and the 2022 tracks respectively. In 2022 despite having poorer initial helpfulness and lower answer prediction accuracy, the overall pipeline still managed to beat our previous best baseline. The result seems to indicate that our reranking that takes in auxiliary features about hosts works well. However, we hypothesize that the poor answer prediction results for TREC 2022 are a result of search topics that require looking at more than the top 16 results.

1.3.1 Thesis Organization

Chapter 2 introduces related work in question answering and previous work for the task of health misinformation. Chapter 3 describes the tasks and the proposed methods for them in greater detail. Chapter 4 describes how the experiments were conducted and discuss the results obtained. Chapter 5 summarizes the work and details future research directions.

Chapter 2

Related Work

In this chapter, we provide a review of research related to the techniques we use throughout this work. We provide an explanation of neural language models, modern state-of-the-art question-answering architectures, and health misinformation on the web.

2.1 Neural language models

The availability of massive amounts of data from the web and advances in computer hardware have brought about significant advances in the field of deep learning. By applying deep learning, researchers in information retrieval and natural language processing have also attained huge gains in a variety of tasks

Early neural natural language processing models relied on shallow networks such as CNNs and LSTMs trained from scratch on domain-specific datasets. While these models used a pre-trained word embedding layer for initial token representations, the rest of the weights would typically be randomly initialized and trained for a specific task. Peters et al. (2018) found that by pre-training entire deep or multi-layer versions of these networks in an unsupervised manner and fine-tuning the models on specific tasks they could achieve much better results than training from scratch in each specific task. This technique of transfer learning had previously proven to be effective in the field of

computer vision. It can help reduce training time and generalization errors in datasets where there are not enough training labels available.

The advent of transformer layers (Vaswani et al., 2017) and the BERT language model (Devlin et al., 2019) which uses multiple self-attention layers stacked on top of each other resulted in another big leap in performance. The significant advantage of stacked self-attention layers is that each token in the input can be processed in parallel, unlike LSTMs (which were the standard for text processing at the time) where to process a token in a sequence the previous token must have finished processing. This means a reduction in training time, which meant models could be trained on even more data.

In the rest of this section, we briefly describe some of the transformer-based language models that are used throughout this work.

BERT

BERT or bidirectional encoder representations from transformers was proposed by Devlin et al. (2019) as a way to use stacked transformer layers for natural language processing tasks. It achieved state-of-the-art performance on various NLP tasks including question answering, sentiment analysis, semantic equivalence, natural language inference, etc. Compared to previous language models which could only process input sequences right to left or left to right, BERT processes input sequences in a bidirectional manner. This was achieved by using a new pre-training task called Masked Language Model. In this task 15% of words were masked and the model was trained to predict the correct tokens to fill in the blanks. This means that BERT can learn the context of a word from both tokens to its right and tokens to its left in a sequence. BERT is also trained for the task of next sentence prediction where consecutive sentence pairs act as positive samples in the training set and random sentences are used as negative samples. The pre-training was done on all of Wikipedia and the Brown corpus.

BERT inputs are prepended with a special $[CLS]$ token and appended with a special $[SEP]$ token. If the input is a pair of sentences a $[SEP]$ token is added between them as well. For classification and regression tasks the $[CLS]$ token's final layer representation vector is fed into a linear layer for classification or regression.

T5

Proposed by Raffel et al. (2020), T5 is another transformer-based language model. Its significant difference from BERT is that it is a text-to-text architecture meaning that any task it is used for must be *cast* to a text-to-text task. For example, to translate a sentence from German to English, the sentence should be prepended with instructions to do so. It was pre-trained on a significantly larger corpus, the Colossal Clean Crawl Corpus (C4 corpus), and fine-tuned on various NLP tasks. When released it achieved state-of-the-art performance on many NLP tasks.

Big Bird

Proposed by Zaheer et al. (2020), Big Bird is a transformer model that uses sparse attention. In regular transformer models to calculate attention scores of tokens, each token in a sequence is compared to every other token at each layer. This means memory usage is quadratic with respect to sequence length, Using Sparse attention reduces the memory requirements of these models and allows for even longer input sequences. This capability makes the model uniquely suited for certain NLP tasks such as question answering and summarization. If we imagine a self-attention matrix where the axes as input tokens, BERT computes attention scores for every element in this matrix, whereas Big Bird only computes attention on the diagonal, sides, and a few random elements. The diagonal and side attention scores give local and global context respectively when encoding a token. Big Bird has better metrics than BERT on natural language tasks and has a limit of 4096 tokens as opposed to 512 tokens. In this work, we will be using Big Bird for document passage extraction but other alternatives exist.

Longformer

Proposed by Beltagy et al. (2020), the Longformer is similar to Big Bird in the way it handles sparse attention. Their implementations vary slightly. Comparing the two across various NLP datasets, Big Bird performs better but requires more computational power.

2.1.1 Reformer

Proposed by Kitaev et al. (2019), the Reformer uses a different approach to reducing the quadratic memory usage of BERT by using locally sensitive hashing (LSH) for computing attention scores. While it is not constrained by an input length limit, it does not scale as well as the previous approaches ($O(L \log L)$ vs $O(L)$).

2.1.2 ConvBERT

Proposed by Jiang et al. (2020), ConvBERT replaces BERT's self-attention with span-based dynamic convolution. Convolution operations can potentially have an advantage over self-attention for capturing local context. ConvBERT has slightly better metrics than BERT but is more efficient, so it could potentially be used in tasks that have longer inputs.

2.2 Question Answering

Question Answering is a fairly old task in the field of information retrieval with the first studies on the subject matter. It first appeared in the Text Retrieval Conference (TREC) in 1999 as the QA track where systems were tasked with retrieving the 5 most probable snippets containing a correct answer from a collection of news articles (Voorhees et al., 1999). This task is an example of the open domain setting where unstructured documents are used as the knowledge source. Over the years TREC has continued to host multiple tracks on question-answering.

Modern question-answering systems architectures can be divided into two components: a retriever and a reader/generator (Zhu et al., 2021). The retriever is a text retrieval system whose goal is to find documents relevant to answering the given question. These retrievers can be either sparse, dense retrievers, or a combination of the two. The reader/generator's goal is to get the correct answer to the question from the retrieved documents. It can either extract a span of text from the documents or generate an answer in a free-form manner.

Sparse Retrievers

Traditionally QA systems used sparse retrievers such as TF-IDF or BM25 (Robertson et al., 1995) in their retrieval stage. These are bag of word ranking functions that estimate the query-document relevance. They rely on term statistics and the exact matching of words, although techniques such as pseudo-relevance feedback can help with vocabulary mismatch. The estimated relevance is proportional to the number of occurrences of query terms in a document with more weight being given to rare words.

In the rest of this section, we discuss QA models that have used traditional sparse models for document retrieval.

DrQA (Chen et al., 2017) is a model which combines traditional sparse retrieval with a neural reader model. This model focuses on factoid questions from Wikipedia. Documents and queries are modeled as bags-of-words and the score of a query document pair is calculated with a variation on TF-IDF as follows:

$$\begin{aligned} \text{tf-idf}(t, d, \mathcal{D}) &= \text{tf}(t, d) \times \text{idf}(t, \mathcal{D}) \\ \text{tf}(t, d) &= \log(1 + \text{freq}(t, d)) \\ \text{idf}(t, \mathcal{D}) &= \log\left(\frac{|\mathcal{D}|}{|d \in \mathcal{D} : t \in d|}\right) \end{aligned}$$

where t is a unigram or bigram. Bigrams can be useful as they take word order into account but they are more sparsely distributed in the collection than unigrams. $freq$ is the number of term occurrences and $\mathcal{D}$ is our document collection from Wikipedia. Wikipedia is a popular collection with many QA papers having used it as a knowledge source.

In Bertserini (Yang et al., 2019a) use the open-source Anserini IR toolkit (Yang et al., 2017) for the retrieval stage. Anserini uses an implementation of BM25 by default and the top 10 documents are retrieved. The authors found that passage retrieval performed better than using either sentence or document retrieval.

In Multi-passage BERT QA (Wang et al., 2019), the authors used the elastic search toolkit [1] for their retrieval. They too found that dividing documents into passages

[1] https://www.elastic.co/

(100-word sliding windows) yielded better performance.

As mentioned sparse retrieval makes use of exact term matching leading to potential problems with vocabulary mismatch. In recent years researchers have looked to neural retrieval methods as an alternative to tackling this challenge.

2.2.1 Dense Retrieval

Constructing low-dimensional representations of text is not a new idea. Methods used for this task included matrix decomposition and shallow neural networks. However, deep learning offers great improvements in representation learning over previous methods. Dense retrieval can be divided into representation and interaction-based retrieval. These two techniques are also called bi-encoding and cross-encoding. One thing to note is that, unlike most sparse retrieval methods, neural dense retrieval can be trained jointly in an end-to-end fashion along with their readers, which means it can be trained in a supervised manner for the specific task.

Representation-based retrieval

Also known as bi-encoder, or dual-encoders, these dense retrieval methods typically follow some variation of the below formula:

$$h_x = E_x(x) \quad h_z = E_z(z)$$

where the query x and document/passage z are fed into language model E and the calculated representation vectors of the two are compared using a dot product. Effectively the query and document are encoded separately (often with a BERT-based transformer) and the similarity score between them is calculated. The most impactful neural retriever-readers such as ORQA, REALM, and DPR have used such a retrieval system. While modern iterations of this retrieval method use BERT-based encoders, the

same architecture has been used with shallow networks for various retrieval tasks as well (Huang et al., 2013) .

One major advantage of this approach is the possibility of pre-computing document representation vectors offline and using fast maximum inner product search (MIPS) to get the top k most similar vectors to a question vector at runtime. There are various ways to implement MIPS, namely Asymmetric LSH (ALSH) (Shrivastava and Li, 2014), date dependant hashing (Andoni and Razenshteyn, 2015) and the most popular approach FAISS (Johnson et al., 2019).

ORQA (Lee et al., 2019) uses two independent BERT-based encoders to encode questions and documents and calculates a relevance score between the two inputs with a dot product. They pre-train their encoders on the inverse cloze task. In the inverse cloze task, the goal of the model is to predict the context of a sentence. In the ICT loss objective, the score of the correct context c of sentences z must be maximized as follows:

$$L_{\text{ICT}} = p_{\text{early}}(z|x) = \frac{\exp(S_{\text{retr}}(z, x))}{\sum_{z' \in \text{BATCH}(\mathcal{Z})} \exp(S_{\text{retr}}(z', x))}$$

To speed up training, the authors use other contexts in the current batch BATCH($\mathcal{Z}$) as negative samples. These other samples are already being processed by the transformer, so by using them we are no longer required to process random samples as negatives

As mentioned, in dense retrieval, document representations can be pre-computed offline. This can also be the case during the fine-tuning phase. In ORQA the authors freeze the document encoder and only fine-tune the question encoder, speeding up the fine-tuning process. The intuition behind this pretraining task is to have representations good enough for evidence retrieval.

Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) uses two independent BERT encoders for retrieval but does away with the inverse cloze task and suggests that training the retriever on question-answer pairs is the better approach. The loss objective here is the negative log-likelihood of the correct passage for the question. It too uses other passages in the same batch as negative samples. Effectively DPR is using supervised training with question-passage pairs while ORQA uses the unsupervised ICT task to train its retriever. The authors also found that including one negative sample

which has a high BM25 score with the question improves retrieval performance. The passage is found by taking a top document returned by BM25 search that does not contain the correct answer.

Instead of document or passage level encoding for later retrieval, researchers have also proposed encoding at the phrase level with DensePI (Seo et al., 2019). By encoding at the phrase level they skip the need for a reader and simply return the top phrase as the answer for the SQuAD-Open dataset (Rajpurkar et al., 2016), achieving better performance than DrQA while being much more efficient.

Interaction-based retrieval

These retrieval models estimate relevance by utilizing token to token interactions instead of a single vector to vector interaction. They are more powerful than representation-based retrievers as they allow for a richer interaction between the tokens of queries and documents. Usually, this entails concatenating the question and document together and using the concatenated text as input for a neural network model. In bidirectional attention flow (BiDAF) (Seo et al., 2017b) a bidirectional LSTM is used for this task. The more modern implementations would use BERT-based transformer models for this task (Pradeep et al., 2021b). The downside of this approach is that it is not possible to do offline document encoding. Thus one forward pass of the model is needed for every potentially relevant document that we need to test. This makes it unpractical to use except for small lists as in a search setting users expect results in a fraction of a second.

By leaving the interaction between question and document tokens to the last level, a model can achieve a higher degree of accuracy while keeping computational costs relatively low. In ColBERT-QA (Khattab et al., 2021), the authors use ColBERT (Khattab and Zaharia, 2020) as the retriever. This retriever uses the same dual encoder architecture with the added step of calculating and summing the cosine similarity of all question and document token vectors. Researchers used have used a final dot-product attention layer for this same purpose as well (Vakili Tahami et al., 2020). SPARTA (Zhao et al., 2021) takes a similar approach where they use a combination of dot products and max-pooling.

Another technique to enhance dense retrieval is using query reformulation where the query is reformulated using an initial set of returned documents. In Generation Augmented Retrieval (GAR) (Mao et al., 2021) researchers use BART (Lewis et al., 2020a) for generating contexts for a given query and append these contexts to the query. For example, they might generate a title for a document and append it to the query. They claim these appended texts better express search intent than just the query by itself. Using these new queries and BM25 they achieved comparable performance to state-of-the-art dense retrieval models. In figure 2.1, we see the general architecture of dense retrieval models.

2.2.2 Reader/Generator

After a QA system has gathered documents relevant to answering a question, it must then use these to predict an answer. It can do so by either extracting spans of text from the documents (reader) or generating text for the answer (generator). Open domain question answering where we find relevant information for predicting an answer is more challenging than the machine reading task where a passage is given along with the question. There is much more information to process and there is the possibility of errors being made in the retrieval stage.

Readers assume that the exact correct answer can be extracted from the documents. They can process retrieved documents independently or jointly.

DrQA (Chen et al., 2017) uses a shallow 3-layer LSTM to process document texts. They use a combination of word embeddings and other features such as: exact matching (whether a document token is present in the query), port-of-speech tags, named entity recognition tags, and term frequency. These document tokens are fed into the LSTM to get their final representations. The representations are combined with the query vector in a bilinear layer and used to predict the start and end tokens for the answer phrase. State-of-the-art models, however, usually use BERT-based transformers for their readers.

BERTserini (Yang et al., 2017) fine-tunes a BERT transformer on the SQuAD dataset. They linearly combine the BM25 and BERT scores of sentences to select the best answer spans.

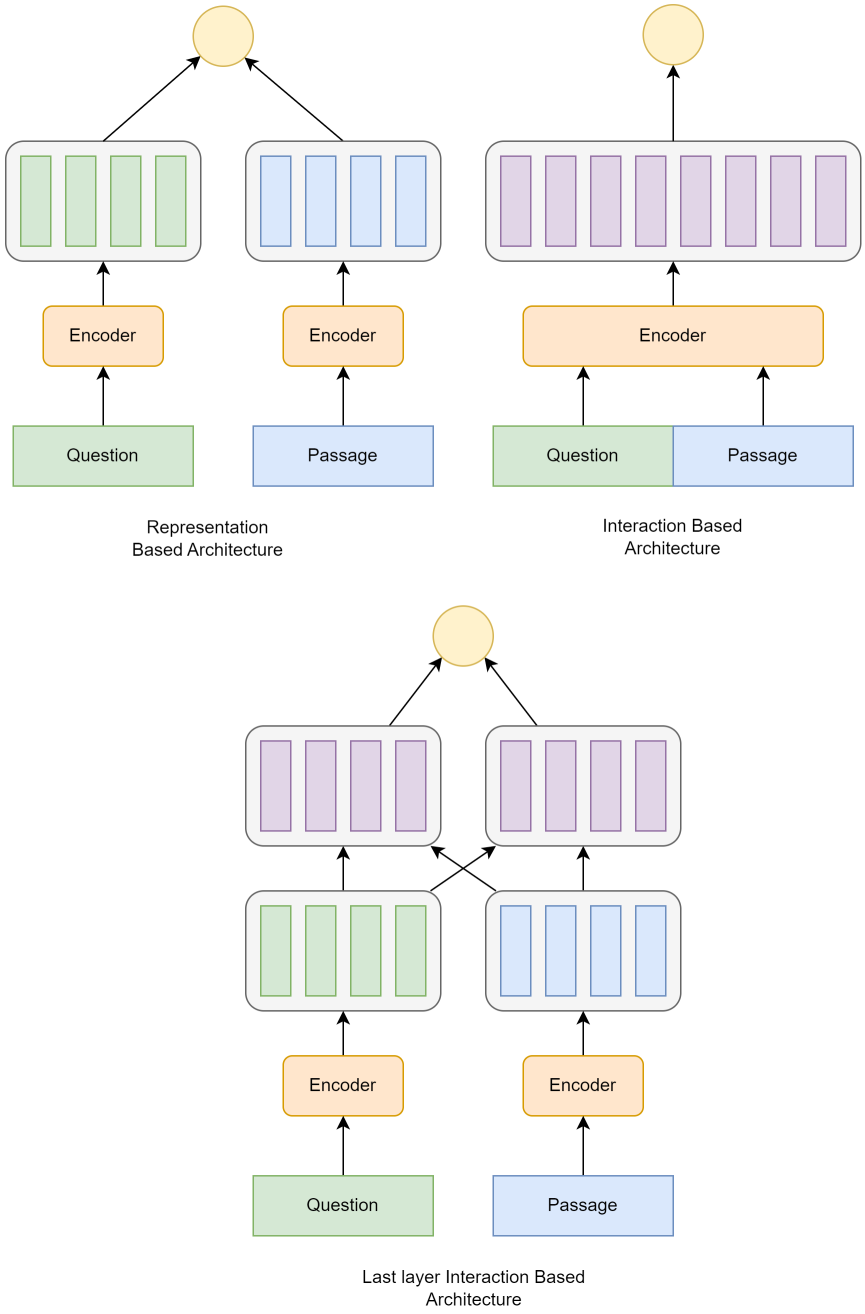

Figure 2.1: Examples of Representation-based, interaction-based architectures as well as an interaction-based architecture which only does token-token comparisons between its two inputs at the last level.

Multi-passage BERT (Wang et al., 2019) feeds token representations into a multi-layer classifier to predict the span start and end tokens. It further enhances this approach by normalizing the scores of tokens across all passages.

By using generators instead of readers models are free to generate free text to answer given questions instead of being limited to selecting start and end tokens in retrieved passages.

Retrieval augmented generation (RAG) (Lewis et al., 2020b) uses the DPR retriever for finding documents. It then concatenates the question with retrieved passages to the question before using BERT to generate an answer.

Fusion in decoder (FiD) (Izacard and Grave, 2021) works similarly to RAG. It uses T5 for encoding passages independently but concatenates the produced vectors together before passing them onto the decoder.

It is also worth noting that larger pre-trained language models are also capable of answering questions in a closed book setting (without any retrieved documents). The 175B parameter version of GPT3 achieved very high accuracy on a variety of QA tasks even in the zero-shot setting (Brown et al., 2020), meaning it could effectively answer questions in these datasets without having ever seen a sample from it.

2.3 Health Misinformation

In this work, we primarily focus on question-answering in the health domain. Search engines are one of the primary ways in which people find health-related information online (Lee et al., 2014). One of the primary challenges in complex question answering on the web is how to aggregate an answer from retrieved documents when they are inconsistent. This is especially prevalent in the health domain. Search engines are a popular tool when it comes to answering health-related questions (Fox and Duggan, 2013). Users searching for medical questions on the web will often face contradictory statements regarding their questions. This became more apparent during the COVID-19 pandemic when we saw a proliferation of harmful and incorrect content. This kind of

misinformation can lead users to make incorrect decisions that can have real-world negative consequences on their well-being.

Various biases can affect the accuracy of a search engine user's decision-making. These biases can come from the content of search results, the search engine, or the user themselves (White and Hassan, 2014). Search engines tend to be biased toward positive outcomes regardless of the truth (White and Hassan, 2014; White, 2014; White and Horvitz, 2015). In these works White and Hassan (2014); White and Horvitz (2015) looked at medical questions that were yes/no questions as well as queries about the efficacy of medical treatments. But, by controlling or mitigating its bias search engines can improve the accuracy of user decision-making. Of particular interest to us are biases stemming from how a search engine displays its results. By biasing the results towards correct or incorrect information and ranking yes answers above or below no answers White (2014) showed that the accuracy of user decision making shifted from 74.9% to 63.1%. In other work, Pogacar et al. (2017) showed that by biasing the rank of the top most correct document authors could alter the accuracy of user-decision making from 23% to 43%.

Thus previous research shows that users cannot reliably make correct medical decisions if we only show them highly relevant documents. In order for them to make correct decisions and avoid potential harm, search engines must accurately determine the correct answer to their questions and bias their search results appropriately.

In another paper, participants were given a series of health questions regarding medical conditions and potential treatments and asked to make decisions on their efficacy based on a search results page. By conducting a think-aloud user study, the authors showed that users will often base their decision on what the majority of results are saying. They also saw a decrease in decision-making accuracy when biasing search results towards incorrect information. During the think-aloud study, users would express their intention of looking for reliable and credible sources. They, however, did not talk about any bias they had towards the topmost ranks or towards web pages claiming treatments were helpful. This indicates that certain biases are at the subconscious level (Ghenai et al., 2020).

Researchers have also looked at click behavior to study this phenomenon (Abualsaud and

Smucker, 2019). They saw that users spending more time viewing incorrect documents are more likely to make incorrect decisions. They also saw that whether the last document was correct/incorrect last document a user viewed had a strong correlation with the user's final decision.

To combat misinformation on the web researchers have proposed a variety of approaches. Epstein et al. (2017) studied the effect of ranking bias in political campaigns. By biasing search results in favor of a candidate they could increase the candidate's vote count in their experiments. However, by alerting the users to this bias using alerts they could reduce the effect this bias had on the outcome.

The TREC health misinformation track focuses on retrieval methods that promote the retrieval of correct and reliable information from the web for health-related decision-making tasks. The track asks participants to retrieve documents relevant to answering a medical question from the world wide web (from the common crawl [2]) and display them to the user while ranking correct credible and informative documents at the top and suppressing incorrect unreliable and uninformative documents towards the bottom.

Researchers in the track have shown that simple intuitive measures such as better retrieval methods, classifying document credibility, and filtering the collection for reliable medical domains can result in modest gains in the helpfulness of retrieved results.

However, Pradeep et al. (2021a) showed that by far the best way to achieve the goal of the task is to simply rerank retrieved documents based on their level of agreement with the correct answer. This approach achieves a much greater reduction of harmfulness compared to other possible solutions (Clarke et al., 2020, 2021a). The TREC health misinformation track has demonstrated over the past few years that relying on features such as the credibility of web pages is sub-optimal compared to simply reranking results based on their agreement with the correct answer. If we know beforehand what the correct answer to a question is, we can rerank a document list based on their level of agreement with the known answer. In reality, we would not have the answer available to us, thus the model must first determine an answer based on the retrieved set of

[2] https://commoncrawl.org/

documents and then do the reranking.

Determining the correct answer to medical consumer questions using modern question-answering techniques and reranking search results so as to boost documents based on their level of agreement is the main task we aim to tackle in this work.

Chapter 3

Methods and Materials

In this chapter, we discuss datasets, newly proposed techniques, and baselines. We begin in section 3.1 by briefly describing the document corpora and datasets used for pretraining. We then describe our task by talking about the TREC Heath misinformation track

3.1 Data

In this section, we describe the datasets and document collections used throughout our experiments.

3.1.1 Clueweb2012

The 2019 TREC health misinformation tack (then called the decision-making track) uses Clueweb12-b13 which is a 7% sample of Clueweb12. Clueweb12 is an English-only web crawl. Clueweb12-b13 contains roughly 50 million documents.

Dataset	documents	tokens	size
C4	365m	156m	305GB
C4 no-clean	1.1B	1.4T	2.3 TB

Table 3.1: Statistics for the C4 dataset. Documents are the text scraped from a URL at a specific time. Tokens are counted using the Spacy (https://spacy.io/) English tokenizer. Size is compressed JSON files (Dodge et al., 2021).

3.1.2 C4

The colossal clean crawled corpus (C4) (Dodge et al., 2021) is a cleaned version of the common crawl. Large language models have in recent years led to impressive gains on many natural language processing tasks. To train large models researchers need large text corpora to pre-train these large models in an unsupervised manner. Naturally, researchers have turned to using web crawls for this task. C4 is such a dataset. It is created by applying a set of filters to a common crawl snapshot. It has been used to train the T5 and Switch language models, which were at the time of their introduction two of the largest neural language models in terms of the number of parameters.

The 2021 and 2022 health misinformation tracks use the C4 collection as their document corpora. They use the uncleaned version which has fewer filters applied to it. The statistics for the C4 document collection are available in table 3.1. The no-clean version contains over 1 billion documents.

3.1.3 MASH-QA

The MASH-QA dataset (Zhu et al., 2020) is a question-answering dataset that was designed to help models tackle questions where answers come from multiple non-consecutive parts of a document. The questions are health-related and the documents are taken from WebMD. The dataset answers are curated by experts as opposed to automatic methods. This ensures the data has less noise and answers to the question can actually be found in the selected spans.

	MASH-QA-S	MASH-QA-M	MASH-QA
# Contexts	5,210	3,999	5,574
# QA pairs	25,289	9,519	34,808
# Train QA	19,989	7,739	27,728
# Dev QA	2,614	879	3,493
# Test QA	2,686	901	3,587

Table 3.2: Statistics for the MASH-QA dataset. MASH-QA-S and MASH-QA-M are the single-span and multi-span subsets of the dataset.

Each sample in the dataset consists of a *(question, context, [answer sentences])* tuple. The natural language question can be answered using one or more sentences from the context which is a long text document. The context will typically be a web document containing multiple paragraphs. The answer sentences are a list of several sentences. They can be from a single span or multiple places across the document. The statistics for the dataset are shown in table 3.2. MASH-QA-S and MASH-QA-M are the single-span and multi-span subsets of the dataset. We use the entire dataset (MASH-QA) in this work. An example from the dataset is displayed in figure 3.1.

The dataset question-answer pairs are consumer health questions sourced from WebMD [1]. The WebMD website consists of a wide variety of healthcare articles. The answers to these consumer health questions have been curated by healthcare experts. Similar to the health misinformation track, finding correct answers to these questions is of great importance as incorrect answers can result in harm coming to users. The answers provided by health experts are taken from these articles with minimal editing. To create this dataset authors match the sentences in the answers to their corresponding sentences in the articles. They do this by first checking for exact matches and then by calculating tf-idf scores and manually matching similar sentences.

Having multi-span answers is the major advantage of this dataset over existing extractive QA datasets. Other multi-span datasets do not use a manual approach to curate their

[1] www.webmd.com

What are tips for managing my bipolar disorder?
Along with seeing your doctor and therapist and taking your medicines, simple daily habits can make a difference. Start with these strategies. *(22 words truncated)* Pay attention to your sleep. This is especially important for people with bipolar disorder... *(178 words truncated)* Eat well. There's no specific diet... *(29 words truncated)* Focus on the basics: Favor fruits, vegetables, lean protein, and whole grains. And cut down on fat, salt, and sugar. Tame stress. *(81 words truncated)* You can also listen to music or spend time with positive people who are good company. *(73 words truncated)* Limit caffeine. It can keep you up at night and possibly affect your mood. *(47 words truncated)* Avoid alcohol and drugs. They can affect how your medications work. *(118 words truncated)*

Figure 3.1: An example from the MASH-QA dataset. Relevant sentences to the question are highlighted.

support documents. As they use automatic techniques such as web search the answers are not guaranteed to be found in the context which can lead to noisy training data.

3.1.4 Topics

For working with the TREC Health misinformation task we use the topics from the 2019 and 2021 tracks. 2020 focused on Covid-19 topics which would not be present in the C4 corpus which is taken from a 2019 crawl. The 2019 and 2021 topics are general-purpose consumer health questions. They consist of a query, which contains topic keywords for a search engine; a description which is a natural language question; a narrative which contains further details regarding the subject matter. The 2019, 2021 and 2022 topics are about medical conditions and potential treatments, thus the narrative would explain these conditions and treatments in greater detail. The narrative can help assessors who are judging documents. The topic also contains a stance or answer as well as a link to the evidence supporting this answer. 2019 and 2021 topics have stances that are helpful or unhelpful, but in this work, we treat the topics as questions and talk about yes/no answers. All topics except three in the 2019 track have descriptions that are yes/no questions where yes is equal to a stance of helpful and no is equal to unhelpful. For those three topics, we reformulate the descriptions as questions. topics from the 2022 track are already in the question answering format. An example topic is in table 3.3. Here the stance for the question is *unhelpful* so in our implementation, we label the answer as a *no*.

We also use a set of consumer health topics from White and Hassan (2014). These were taken from reviews from the Cochrane library [2]. Cochrane is a charitable organization with the aim of organizing medical research. The group conducts systematic reviews of healthcare interventions and diagnostic tests and publishes them in the library. Some of the health misinformation track topics are also based on Cochrane reviews. Overlapping topics were removed to give a balanced topic set with 45 yes answers and 45 no answers. This topic set has been proven effective as training data for the health misinformation

[2] cochrange.org

Topic	104
Query	duct tape warts
Description	Does duct tape work for wart removal?
Narrative	Duct tape is a plastic and cloth backed adhesive tape commonly available and known to be useful for quick repairs. Warts are skin growths caused by a viral infection. A very useful document will discuss the effectiveness of applying duct tape to warts for their removal. A useful document would help a user decide if duct tape is an effective remedy for warts by providing information on recommended methods to treat warts, and may or may not mention the use of duct tape for this purpose, but which do not directly address the effectiveness of duct tape for wart removal.
Stance	Unhelpful
Evidence	https://pubmed.ncbi.nlm.nih.gov/22972052/

Table 3.3: An example topic from the health misinformation track.

track in Zhang et al. (2022) and is available in the projects GitHub repository [3]. We will refer to this topic set as the White and Hassan topics in future sections.

3.2 Tasks

In this section, we describe and formalize the various tasks used in the health misinformation pipeline. The question-answering pipeline proposed in this work is designed for the TREC health misinformation tracks. We use the 2019 and 2021 tracks.

In 2019 and 2021, the topics (questions) are regarding a medical condition and a potential treatment. These pairs are labeled as unhelpful, helpful, or inconclusive. Unhelpful means that the treatment does not help the condition. Helpful means the treatment helps the condition and Inconclusive means that there is not enough evidence to support either claim. Only the 2019 track contains inconclusive topics. In the 2022

[3] https://github.com/UWaterlooIR/golden-gaze

track, the topics are more general medical questions but their labels are still binary yes/nos. These topics are labeled by the track organizers based on reviews done by medical professionals from trusted sources.

For each track, participants are tasked with creating models that rank the documents to boost correct and suppress incorrect documents. Runs can use the helpfulness labels provided by the track organizers but these would count as a non-automatic run. These runs are pooled together and given to assessors for judgments. The assessors judge documents based on their stance (helpful/unhelpful or yes/no), relevance, and credibility. Statistics on these judgments are available in 3.5.

In total there are 151 topics across the three years. Topics are comprised of a unique id, a query that contains keywords to be used for search, a description that takes the form of a natural language question, a narrative that gives a more detailed account of the medical terms used in the question, and the search intent of the topic's creator.

From 2021 onward the track no longer has inconclusive topics. For consistency across tracks, we disregard the inconclusive topics from the 2019 topics.

In 2021 the track uses graded scores and the compatibility metric (Clarke et al., 2021b) for evaluation. If a document's stance matched the correct answer it would have a higher grade depending on its credibility and informativeness. If it disagreed with the correct label then its score would be lower if it had higher credibility or informativeness.

The compatibility of a ranking S is measured by calculating the rank biased overlap between the ranking and an ideal ranking T as such

$$RBO(S, T, p) = (1 - p) \sum_{d=1}^{\infty} p^{d-1} A_d$$

where A_d is the *agreement* of proportion of S and T that are overlapped at depth d. p is a searcher persistence. It effectively gives a higher weight to the higher ranks. It can be thought of intuitively as the probability that a search engine user will continue to the next item at each item. Ranked biased overlap calculates the similarity of two lists. If the

Judgments	Score
very useful, correct, very credible	12
useful, correct, very credible	11
very useful, correct, credible	10
useful, correct, credible	9
very useful, correct, not credible or not judged	8
useful, correct, not credible, or not judged	7
very useful, neutral or not judged, very credible	6
useful, neutral or not judged, very credible	5
very useful, neutral or not judged, credible	4
useful, neutral or not judged, credible	3
very useful, neutral or not judged, not credible or not judged	2
useful, neutral or not judged, not credible or not judged	1
not useful, not judged, not judged	0
very useful or useful, incorrect, not credible or not judged	-1
very useful or useful, incorrect, credible	-2
very useful or useful, incorrect, very credible	-3

Table 3.4: Grade scores used for calculating compatibility in the 2021 health misinformation track. An ideal ranking must place all higher scored documents before any lower scored document.

	2019 qrels	2021 qrels
Number of judgments	22,859	12,778
Number of useful documents	3,137	4,155
Number of very useful documents	1,028	2,314
Number of supportive documents	3,025	3,667
Number of dissuasive documents	161	889
Number of credible documents	2,229	3,339
Number of very credible documents	-	652
Maximum number of words	61,511	27,535
Minimum number of words	65	47
Average number of words	1,543.345	2,746.622

Table 3.5: Statistics for the assessor judgments made for the tracks 2019, 2021, and 2022

lists are identical it returns a value of 1 and if they are completely different it returns a value of zero.

In the 2021 track, two rankings helpful and harmful are created for each topic. A helpful ranking represents the best-case scenario where documents are ranked by the scores in table 3.4. In the harmful ranking, the worst documents are placed on top. To evaluate a ranking, helpful and harmful compatibility are calculated and subtracted from one another as follows:

$$\text{Compatibility}_{\Delta} = \text{Compatibility}_{\text{helpfulness}} - \text{Compatibility}_{\text{harmfulness}}$$

Basically runs must rank documents with a higher score first to achieve a high compatibility score.

Typically runs in TREC contain 1000 documents, but by using a default p of 0.95, we effectively limit the evaluation to the top 20 documents.

In the 2022 track in addition to graded relevance, assessors were tasked with applying preference ordering to the judgements. They used an interface in which they were shown two of the top retrieved documents and were tasked with specifying which was the more

useful and correct document. Using these preference judgments the top scoring documents would be reranked in the "ideal" ordering used when calculating compatibility.

3.3 Methods

In this section, we describe our proposed QA pipeline designed to tackle the health misinformation track. But before that, we first propose a simple domain filtering approach that, while not very effective, can yield some insight into the problem at hand.

3.3.1 Domain Filtering

One seemingly obvious solution to the task is to simply filter the document collection to contain only hosts/domains that are credible. One possible solution to determining credibility is to use already available credibility certifications. Here we used the HONCode certification. These certifications are provided by the Health on the Net Foundation which was a non-profit organization [4]. The organization's website provided health information and issued the certificate to compliant websites that requested it. Compliance is based on a code of conduct that aimed to promote useful and reliable medical information on the web. The code of conduct mentions factors such as transparency of authorship and sponsorship, authority (advice given by medical professionals), and attribution (references to sources and their dates). The code of conduct taken from the website is provided below [5]:

- Authority – Any medical or health advice provided and hosted on this site will only be given by medically trained and qualified professionals unless a clear statement is made that a piece of advice offered is from a non-medically qualified individual or organization.

[4] https://myhon.ch/en

[5] http://web.archive.org/web/20190808172632/http://www.hon.ch/HONcode/Conduct.html

- Complementarity – The information provided on this site is designed to support, not replace, the relationship that exists between a patient/site visitor and his/her existing physician.

- Privacy – Confidentiality of data relating to individual patients and visitors to a medical/health Web site, including their identity, is respected by this Web site. The Web site owners undertake to honor or exceed the legal requirements of medical/health information privacy that apply in the country and state where the Web site and mirror sites are located.

- Attribution – Where appropriate, the information contained on this site will be supported by clear references to source data and, where possible, have specific HTML links to that data. The date when a clinical page was last modified will be clearly displayed (e.g. at the bottom of the page).

- Justifiability – Any claims relating to the benefits/performance of a specific treatment, commercial product, or service will be supported by appropriate, balanced evidence in the manner outlined above in the attribution principle.

- Transparency of authorship – The designers of this Web site will seek to provide information in the clearest possible manner and provide contact addresses for visitors that seek further information or support. The Webmaster will display his/her E-mail address clearly throughout the Web site.

- Financial disclosure – Support for this Web site will be clearly identified, including the identities of commercial and non-commercial organizations that have contributed funding, services, or material for the site.

- Advertising policy - If advertising is a source of funding it will be clearly stated. A brief description of the advertising policy adopted by the Web site owners will be displayed on the site. Advertising and other promotional material will be presented to viewers in a manner and context that facilitates differentiation between it and the original material created by the institution operating the site.

There are approximately 8000 hosts in the collection that have HONCode certifications. Previous research has shown that certification is a reliable indicator of quality health information and most uncertified websites do not adhere to the HONCode principles (Laversin et al., 2011). Such a few numbers of hosts are most likely not able to answer general-purpose medical questions across a wide set of domains. To this end, we proposed expanding the collection by exploiting the hostname link graph[6].

Common crawl provides this link graph which details how hosts or domains are connected to each other via links. In this graph, nodes are hostnames and edges are the presence of a link between the contents of the two nodes. We hypothesized that websites with the HONcode certification are more likely to link to other reliable websites even if they do not have the certification.

There are approximately 4 million nodes and 4 billion edges in the host link graph. Our goal is to find reliable hosts such as medical journals and organizations that do not have the HONCode certification. Most likely because they have never applied for receiving it. We use a variation of topic-sensitive PageRank (Haveliwala, 2002) in an attempt to find these hosts. We create a subset of the link graph as follows: For nodes, we take all the 8000 domains and all the domains they link to. For edges, we take all edges where the source is one of our 8000 reliable domains. We calculate PageRank scores for all nodes but only randomly jump to reliable domains. We then take only the top 10000 highest-scoring nodes. We end up with 10000 domains and roughly 30 million documents. However many of these documents are non-medical and irrelevant to the task at hand. These domains are typically hub pages that end up with high PageRank scores due to the sheer number of incoming links. In the next step, we will filter these out.

To this end, we train a medical text classifier which we then use to tell us what proportion of pages for each given host is actual medical content. To train this classifier we use the 2019 TREC health misinformation topics on the ClueWeb12 collection. We find the top 100 pseudo-relevant documents for each topic by taking documents with the top 100 BM25 scores. We use these as positive samples. For negative samples, we randomly select websites from common-crawl minus documents with hosts in the 8000

[6] https://commoncrawl.org/2019/02/host-and-domain-level-web-graphs-nov-dec-2018-jan-2019/

reliable domain list. Given the size of the crawl, it is unlikely that our random samples will contain medical documents. We train a model using a linear support vector machine model and validate with 5-fold cross-validation where we train on 40 topics and test with 10 in each fold. We use a simple tokenization scheme where we remove all special characters, punctuation and single characters and split on white space and transform documents int TF-IDF vectors. The TF-IDF implementation is also very simple using linear term counts:

$$\text{TF-IDF}(w, d) = freq_{w,d} \cdot \frac{N}{freq_{w,C}}$$

where w is the word feature, d is the document and C is the entire collection. As the task is very simple (distinguishing medical from non-medical text) the model achieves a very high accuracy and F1 score of 0.99 and 0.98 with a threshold of 0.5 for the binary classification task.

With this classifier, we filter out documents whose text is classified as non-medical and are left with 1,829,111 documents. Figure 3.2 shows an example of how this process works. A reputable website will have many incoming links from HONCode certified medical domains such as webmd.com. So non-credible websites such as junk-medicine.com will have low PageRank scores meaning all of its web documents will be removed from the collection. Reputable non-medical websites such as google.com will still have high PageRank scores. After running the document classifier non-medical documents will be removed. Thus leaving non-certified but credible medical documents like those from bmj.com for example.

In table 3.6 we see the top 5 domains along with their PageRank scores after filtering out irrelevant domains. The top domains are generally well-established and reliable sources of medical information.

3.4 Question Answering pipeline

In this section, we provide a detailed overview of the question-answering pipeline used for the TREC health misinformation track. The pipeline's overall architecture can be seen in

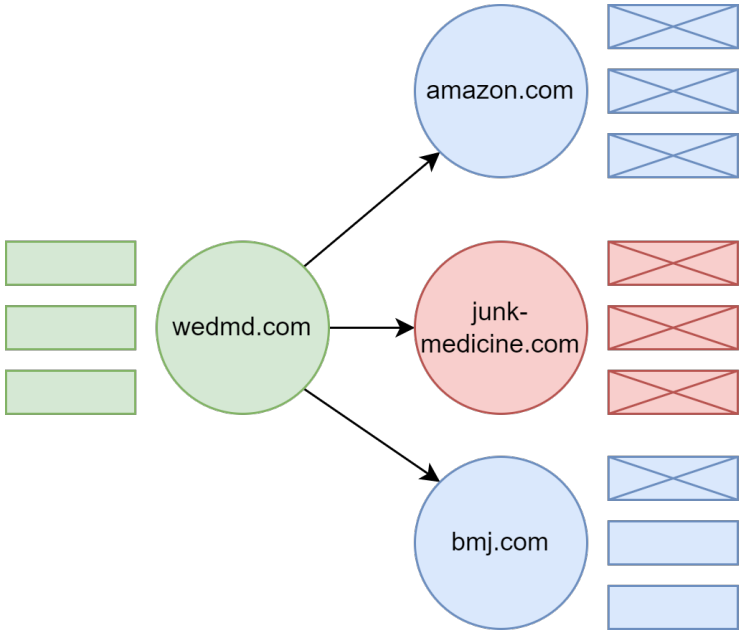

Figure 3.2: A subset from the host link graph. In our domain filtering approach, non-credible websites and their documents are removed, such as those from junk-medicine.com. Nonmedical documents from the remaining websites are removed using a text classifier.

Hostname	PageRank Score
gov.fda	74.077
gov.clinicaltrials	55.691
gov.nih	40.851
gov.medlineplus	35.907
org.cancer	33.679
com.webmd	25.860

Table 3.6: Pagerank scores for the top 5 hosts.

figure 3.3. Given a query, the system retrieves a set of initial documents using a sparse retrieval method. We use Anserini BM25 with default parameters. This initial retrieval is being done on the entire collection. In settings with a large initial collection, most retrieval pipelines will retrieve using a fast simple technique before reranking using more complicated approaches. This makes sense from a practical standpoint but it also means that the final retrieval quality is heavily dependent on the quality of this initial retrieval.

The next step in the pipeline is to employ a dense retrieval algorithm to rerank this initial set of documents. This is to move more relevant documents to the top of the list. Later on in the pipeline, we will be using top documents to predict an answer to the given question. However as web documents are fairly long, the system needs to employ a passage extraction mechanism. This will shorten document lengths to be more manageable, as most neural language models do not work with long input sequences. Transformer models run into memory limitations as they process tokens in parallel. In general any neural network will run into issues with long input. LSTMs for instance suffer from vanishing gradients. CNNs will have trouble modeling the global context.

We now have a small set of potentially relevant documents we can work with. At this stage, the QA system must rerank the list of documents to push documents that agree with the correct answer to the top and suppress those that disagree towards the bottom. But to do this it must first determine what the correct answer is. Therefore the system must aggregate the stances of its ranked list. The system must determine what each web document's answer to the given question is, and then it must apply a weighted aggregation to determine what the final answer is. The exact mechanism of the weighted aggregation can vary but it can take into account the textual content of web pages and also auxiliary features such as the hostname, PageRank scores, credibility labels, etc.

Once the system has an answer to the given question it can then perform a soft reranking of the document list based on its prediction of each document's answer, boosting those in agreement and suppressing those without. The reranking is more aggressive the more confident the answer is in its prediction final answer and the more confident it is about the stance of the document being reranked.

In the following section, we will discuss pipeline components in greater detail.

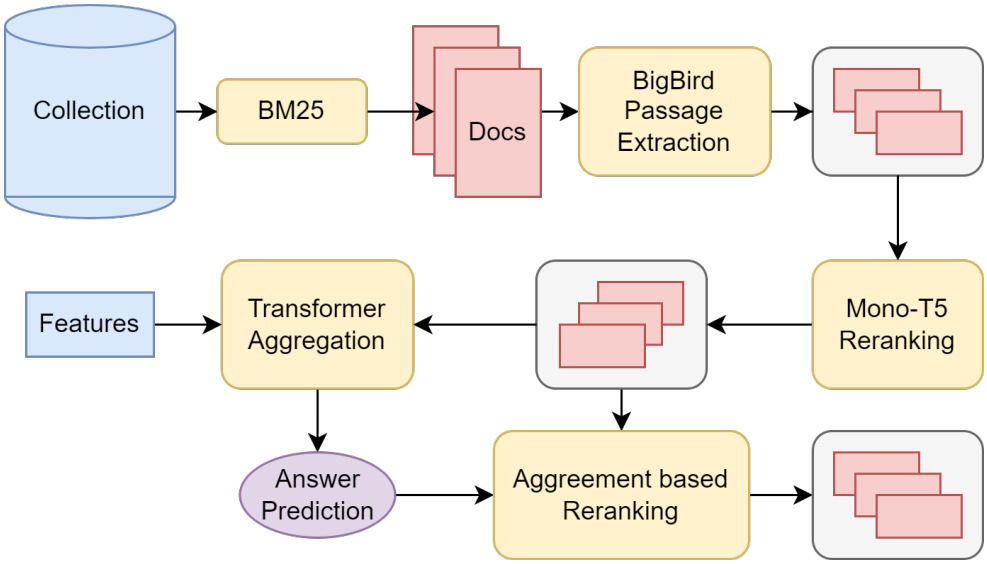

Figure 3.3: Our proposed pipeline. The stages consist of 1. Initial retrieval with BM25, 2. Passage extraction with the Big Bird transformer, 3. Neural reranking with Mono-T5, 4. Answer aggregation, and 4. Soft reranking.

3.5 Passage Extraction

One of the key components of this pipeline is the method of passage extraction. For more accurate question answering, systems tend to use BERT-based transformers. However, web documents are typically much larger than the typical 512 token limit of most transformer models.

Pradeep et al. (2021a) use a fairly common approach to passage extraction. A web document is divided into sliding windows of 6 sentences and steps of 4 sentences. Previous research (Liu and Croft, 2002) has shown that passages can be effective representations of a document for use in retrieval. Each passage is independently fed into the Mono-T5 (Pradeep et al., 2021b) transformer model. After getting the relevance scores for each of these passages from Mono-T5, the top-scoring passage is selected as the extracted passage for the document. Mono-T5 is a version of the T5 transformer that has been pre-trained on the MS-MARCO (Nguyen et al., 2016) dataset for relevance ranking.

In practice, there are two issues faced by this method. Firstly it is computationally expensive. The mean number of sentences for the C4 collection of Web documents is roughly 32. For a document of this length, using the method in Pradeep et al. (2021a), we would have to run a transformer model 29 times to get an extracted passage. Secondly, relevant information to a query may be spread out across a web page in nonconsecutive spans. Especially when it comes to medical questions as shown in the MASH-QA dataset (Zhu et al., 2020), where answers curated by experts from medical articles for consumer health questions use non-continuous spans of text from those articles. In general, to reach a decision, humans or models will sometimes need to piece together information from multiple parts of a document. Relying on single-span extraction techniques could mean our passages may not contain all relevant information from a document.

Another method is to use domain-specific heuristics to extract sentences from a document. Zhang et al. (2022) in the TREC 2021 health misinformation track used specific keywords to determine whether a sentence should appear in the extraction. The queries in that year's track are regarding medical treatments and illnesses, thus certain specific words can be good indicators of relevance. They score each sentence based on the presence of words such as "dangerous", "effective", etc. as well as the query terms.

Our proposed method is a generalization of this approach. As mentioned, web documents tend to be long consisting of multiple sentences. Typically too long to fit within the 512 token limit common in transformer models. Answers to questions can also tend multiple non-consecutive sentences. To train a model for this purpose we use the MASH-QA dataset (Zhu et al., 2020). The MASH-QA dataset is specifically designed to tackle this problem. It is a question-answering dataset built from the medical domain where answers to questions need to be extracted from multiple non-consecutive parts spanning across a document. Its documents consist of multiple paragraphs, with relevant spans of texts spanning the entire document.

Our proposed passage extraction model works as follows: Using Spacy [7] we split the document into sentences and place a special [SEN] at the end of every sentence. We prepend the question to the document, pacing the [CLS] token at the beginning and

[7] https://spacy.io/

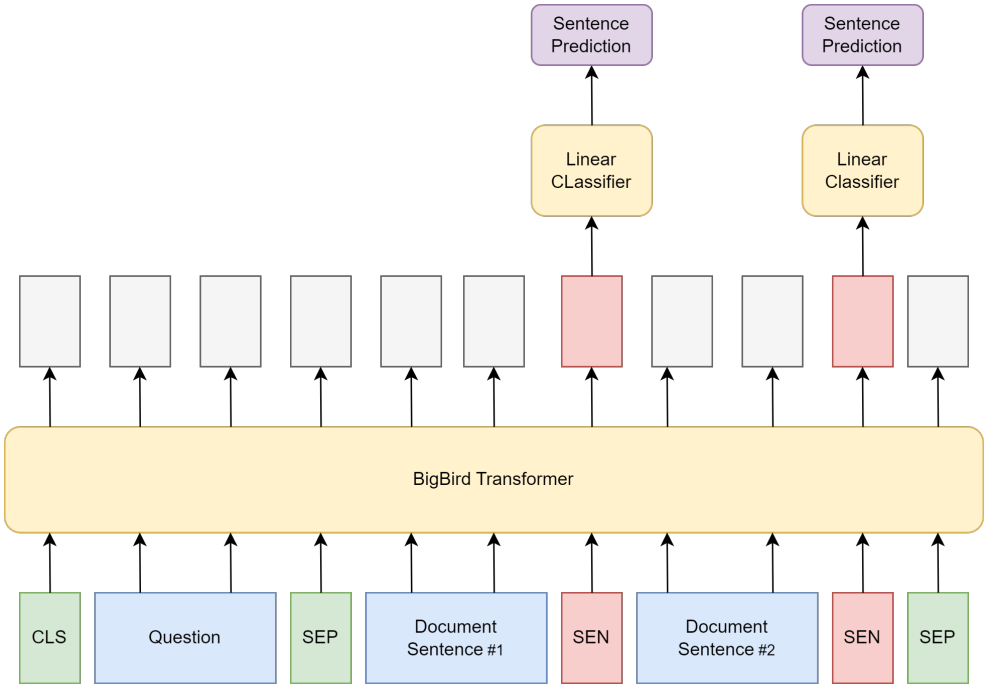

Figure 3.4: Query-biased passage extraction using the Big Bird transformer and special [SEN] tokens.

placing [SEP] before and after the document. We feed the concatenated text into a Big Bird transformer model. We take the final layer representation vectors for the [SEP] tokens and classify them as relevant or non-relevant with a linear classifier. This method is superior to classifying sentences individually as it takes the entire document context into account and performs better while having a simpler architecture compared to existing multi-span passage extractors. The process is displayed in figure 3.4. The threshold for including and not including sentences can also be adjusted, giving either higher precision or higher recall depending on what is more important for the task at hand.

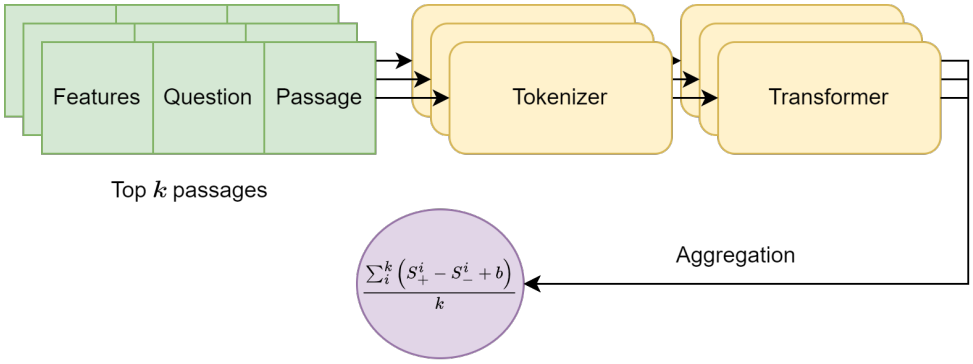

Figure 3.5: Query-biased passage extraction using the Big Bird transformer and special [SEN] tokens.

3.6 Aggregation

In the aggregation stage, the model must predict a correct answer for the given question. Previous work (Zhang et al., 2022) has shown that it is possible to predict an answer using stance detection and hostnames. In this work, we proposed a generalization of this approach. To do this, we proposed aggregating the top-ranked document passage answers to the given question. Once we have extracted a query-dependant passage from the documents, we rank using a neural retrieval method (Mono-T5) to ensure the top documents are relevant. At this stage, we use a BERT-based question-answering model to predict each document's answer to each individual question. Previous research has shown that weighting features need to be taken into account. The advantage of this approach is that it takes language features into account, meaning the aggregation model can learn to weight documents that use credible and non-credible language differently. This approach can take advantage of auxiliary features as well. The overall procedure is displayed in figure 3.5.

The exact process is as follows. For the top k passages, we extract auxiliary features. We concatenate the question and the passage for each of these. We use hostname and HONcode certifications as features. We prepend the hostname as plain text to the question-passage pair. We use a special token ([HON]) for HONCode certifications and prepend the token to the question-document pairs that have a hostname with a

HONCode certification. We feed the resulting text to a BERT transformer sequence classifier. The transformer outputs three scores for yes/no/neutral. We average the scores of the top documents as follows:

$$Pred = \frac{\sum_i^k S_+ - S_-}{k} + b$$

where the bias b is a learned parameter. This prediction can be used to rerank all of the documents based on their level of agreement.

3.7 Reranking

At the final stage, we need to rerank all of the documents to make sure documents agreeing with the predicted answer float to the top of the tanked list and those that are in disagreement are suppressed.

We concatenate the question with passages of all documents in our retrieved set. We do not add any features this time. We feed the text into a BERT transformer to get a yes/no/neutral prediction. We calculate the prediction score as $P_{qd} = S_+ - S_-$. Once we have a prediction for each document we adjust their Mono-T5 scores as follows:

$$S'_{qd} = \frac{2S_{qd}}{(1 + e^{\alpha \cdot P_q \cdot P_{qd}})}$$

where $S_{qd} \in (0, \text{inf})$ is the Mono-T5 score for document d's passage and a question q, P_q is the aggregated question answer prediction, P_{qd} is the answer prediction for a documents answer and α is a hyper-parameter. For a document that has a very high level of agreement with the topic prediction, the formula will double its Mono-T5 score. For documents with a low level of agreement, the score will be zero. The intuition behind the formula comes from the notion that documents in disagreement with the correct answer should not appear at the top at all no matter how credible. However, a non-credible document that agrees with the correct answer should not be placed higher than a credible

document that also agrees with the correct answer simply due to having a higher degree of agreement.

In the next chapter, we will show evaluations for each of the proposed methods.

Chapter 4

Results

In this chapter, we discuss obtained results from the methods proposed in the previous chapters. We will first discuss the effectiveness of domain filtering on the TREC 2021 Health misinformation track. Then in section 4.2, we will look at the performance of our passage extraction method on the original train-test dataset (MASH-QA). In section 4.3, we will look at how well the proposed aggregation method works when predicting answers for health questions. Finally, in section 4.4, we look at how well our reranking pipeline works when reranking an initial set of retrieved documents for displaying on a search engine results page.

4.1 Domain Filtering

In this section, we look at the effectiveness of filtering domains to include only credible websites before running a simple BM25 retrieval function.

Looking at a subset of the automatic runs from the 2021 Health Misinformation Track in table 4.1, we see that the document collection to only contain credible documents results in a fairly significant boost to helpfulness. We see a reduction in harmfulness from 0.144 to 0.119 as well as a boost to helpfulness from 0.122 to 0.147. These results are from when we filter out the collection to contain only hosts in the HONCode certification. When we

apply our proposed expansion to the filtering collection we see a slightly bigger boost to helpfulness up to 0.164. In section 3.3.1, we discussed how in this approach we expand our filtered collection by running a topic-sensitive PageRank on a subset of the common crawl link graph that only includes HONCode domains and domains they link to.

We see that by simply filtering the collection and running a basic BM25 algorithm, we get a similar compatibility score to state-of-the-art retrieval models that use deep learning on the whole collection. Meaning there is some value in detecting and filtering out non-credible websites. However, we also see that filtered collections can limit the number of helpful documents found even if they reduce the number of harmful documents.

By running a deep learning retrieval model Mono-T5 on our filtered-expanded collection subset we see a boost in performance compared to BM25. However, we fail to reach the compatibility of the same method on the unfiltered collection. This is a strong indicator that filtering the collection does not work for all topics and is not a useful approach for general medical question answering. Mono-T5 finds much more helpful content in the entire collection at the cost of finding more harmful documents.

We can also take a look at per topic performance in table 4.2 where we see that collection filtering has a negative performance on certain topics. While there is no single explanation as to why certain topics do better or worse. An analysis of results tells us that for topics where reputable sources are unlikely to mention them such as "copper bracelets reduce pain", it can be hard to find helpful documents.

The conclusion to our analysis of filtering the collection is that it is too impractical to filter based on the quality of hosts. Since our goal is to build a system that can answer a wide variety of medical questions from the web, we need to look at a more effective solution.

4.2 Passage Extraction

In this section, we look at the performance of our proposed passage extraction method on the original training set before moving on to the downstream task in the next sections.

Collection	Retrieval Method	Helpfulness	Harmfulness	Delta
Unfiltered	BM25	0.122	0.144	-0.022
	Mono-T5	0.170	0.119	0.051
Filtered	BM25	0.147	0.119	0.027
Filtered + Expansion	BM25	0.164	0.123	0.040
	Mono-T5	0.151	0.112	0.039

Table 4.1: Compatibility metrics for the 2021 topics in the C4 collection. Unfiltered is on all of C4. Filtered is for only hosts with a HONCode certification. filtered + expanded is for HONcode hosts and reliable health-related hosts that they link to.

We train and evaluate on the MASH-QA dataset. As explained in 3.1.3, the MASH-QA dataset is specifically designed to train models for extracting non-continuous spans. Much of the work on extractive question answering is focused on extracting continuous spans containing the answer to a question. For certain domains or tasks, this may not be ideal as answers are not short and require more context to be answered properly. And this required context could be spread across multiple points in a document

A naive approach to this kind of passage extraction is to simply classify each sentence in a document as being relevant or not to a query. The authors show that this approach performs poorly. We redo the same experiment as those authors and report the numbers in table 4.3.

We use a BERT sequence classifier where the question and sentence are concatenated together. Each question and sentence pair are then classified as being relevant or not. This naive approach tends to perform poorly. Zhu et al. (2020) claim this is due to the model lacking the greater context when it comes to deciding the relevance of a sentence.

For MASH-QA, the authors propose a new passage extraction model called Multi-Co. The model aims to tackle the shortcomings of the sentence classifier approach. The model concatenates the question and document before feeding the concatenated text into an XLNet (Yang et al., 2019b) transformer model. XLNet is based on Transformer-XL (Dai et al., 2019) which is designed to work with longer documents. They do this so they

Topic	Query	Unfiltered	Filtered+expanded	Delta
101	ankle brace achilles tendonitis	-0.200	-0.112	0.089
102	tepid sponge bath reduce fever children	0.016	-0.109	-0.124
103	folic acid dementia	0.098	0.063	-0.035
104	duct tape warts	0.006	0.030	0.024
105	put ice on a burn	0.068	0.085	0.017
106	vitamin b12 sun exposure vitiligo	0.134	0.334	0.200
107	yoga asthma	0.097	0.145	0.048
108	starve a fever, feed a cold	0.128	0.035	-0.093
109	selenium cancer	0.257	-0.027	-0.284
110	birth control pill ovarian cysts treatment	-0.137	-0.001	0.137
111	zinc supplements pregnancy	-0.180	-0.204	-0.023
112	evening primrose oil eczema	-0.075	-0.247	-0.172
114	vitamin e cream for skin scars	0.104	0.132	0.029
115	magnesium migraine prevention	0.096	0.048	-0.049
117	fermented milk blood pressure	0.097	-0.004	-0.101
118	dupixent eczema	0.186	0.136	-0.050
120	imitrex migraine	0.178	0.039	-0.139
121	light therapy lamp depression	0.091	0.122	0.031
122	aleve migraine	-0.034	-0.157	-0.123
128	steam shower croup	-0.217	-0.112	0.105
129	minoxidil balding hair growth	-0.017	0.063	0.080
131	l-theanine supplements anxiety	0.004	0.083	0.080
132	inhaling steam common cold	-0.292	-0.351	-0.059
134	remove tick with vaseline	0.099	-0.061	-0.160
136	dates iron deficiency anemia	0.112	0.609	0.496
137	vinegar fish bone stuck	-0.104	-0.453	-0.348
139	copper bracelets reduce pain	-0.044	-0.238	-0.194
140	fungal cream athlete's foot	0.026	0.305	0.280
143	tylenol osteoarthritis	-0.050	0.112	0.161
144	music therapy depression	0.207	0.549	0.343
146	vitamin d asthma attacks	0.542	0.604	0.061
149	hip osteoarthritis at-home exercises	0.050	0.223	0.172

Table 4.2: Comparision of compatibility metrics for Mono-T5 runs on 2021 topics for the c4 collection with and without filtering.

can encode larger contexts as typical transformer models like BERT have a 512 token limit. After obtaining XLNet token representations, the model then applies self-attention within each sentence to get a fixed dimensional vector for each sentence. It appends [CLS] to these new sentence representation vectors. The model then applies inter-sentence self-attention between the sentence vectors before feeding the resulting vectors into a linear classifier. The general idea behind this model is to modify the sentence vectors being fed into the relevance classifiers so that they include information from elsewhere in the context.

As discussed in detail in section 3.5, the model we propose for this task is much simpler while achieving better results. Instead of XLNet, we use the Big Bird transformer model. We append a special token to the end of every sentence and feed this special tokens representation vector into a linear classifier. The main difference between our model and Multi-Co is that we show that there is no need for a hierarchical self-attention layer for getting sentence representations and that we can rely on the already very powerful transformer models to create these sentence representations for us.

We compare the performance of passage extraction for the MASH-QA dataset in table 4.3. We include data from the original paper as well as our own experiments. We use sentence-level precision, recall, and F1 macro metrics for evaluation. These metrics will reward partially correct answers and take class imbalance into account as the majority of sentences in a document will not be relevant for answering a given question. Using the newer and larger Big Bird model and swapping the task-specific architecture for special sentence tokens gives a sizeable increase in F1 to 74.37 compared to Multi-co's 57.00. We see that recall is higher than the precision meaning we can find more sentences but precision is lower meaning some sentences are being mistakenly classified as relevant. In the next section, we will see what effect the classification threshold has on our downstream task.

4.2.1 Implementation Details

We use the base version of Big Bird, while MASH-QA authors use large versions of transformers. We train our Big Bird model for 3 epochs with a learning rate of 2×10^{-5}

Model	Precision	Recall	F1
BERT	56.8	16.42	25.44
RoBERTa	56.18	16.25	25.21
XLNet	57.70	19.06	28.65
Multico	58.16	55.90	57.00
BioBERT	18.67	76.67	30.03
Big Bird	70.57	81.21	74.37

Table 4.3: Comparison of sentence level metrics for multi-span passage extraction on the MASH-QA dataset. Individually classifying sentences as relevant to queries has very poor performance. Models that take the entire document context into account are much better for this task.

and a weight decay of 0.01. We use a max input size of 2048. We use a batch size of 2. While Big Bird's sparse attention is less memory intensive than typical BERT models the size of the model and input mean we are limited to a batch size of 2 on a 40GB GPU. The epoch with the best F1 on the validation set is saved and used to evaluate the test set. The 2048 limit is sufficiently long for the MASH-QA dataset but for inference on downstream tasks, we would need to split a document to classify all the sentences.

4.2.2 Efficiency

Our proposed passage extraction method requires less computation than Mono-T5 passage extraction. An NVidia A100 can run roughly 60 samples a second through a Big Bird transformer and 460 samples a second through a T5 transformer (Both using their base size setting). Mono-T5 has an input limit of 512 and Big Bird has an input limit of 4096. Our collection documents have a mean length of 32 sentences. Mono-T5 operates on sliding windows of size 6 with a step of 3. This means that in practice the Mono-T5 will be slower in the majority of cases. As an alternative to Big Bird, we could experiment with Longformer (Beltagy et al., 2020) which can still long documents but has a compute cost closer to T5. However, we leave this for future work.

4.3 Question Answering

In this section, we analyze the effectiveness of our proposed question answering approach. As discussed in greater detail in section 3.6, after reranking extracted passages with a dense neural retrieval model (Mono-T5), we take the top 16 passages, prepend them with the question and manual features before aggregating their outputs to get a final answer. We chose 16 passages as we are limited by GPU memory.

We train on the White and Hassan topics and use the 2019 and 2021 topics as development and test sets. We train for 12 epochs and save the model with the highest accuracy score on the development set and use that model to evaluate the test set. We use the Adam optimizer with a learning rate 0f 1×10^{-5} for the transformer model and a learning rate of 1×10^{-3} for the added bias in the final classification layer. The bias is initialized as 1×10^{-5}. To initialize transformer parameters we use the weights from BioBert (Lee et al., 2020) that we further fine-tuned on the PubMedQA dataset.

The hyper-parameter sweep on the development sets is displayed in table 4.4. The threshold is the minimum score required for a sentence to be included in the extracted passage. Features are defined as HH meaning both HONCode and Hostnames were used as features, H meaning only hostnames were used and N meaning no auxiliary features were used. The addition of features appears to have little impact on the accuracy of the 2021 topics. But both features appear to help increase the accuracy of the 2019 topics. The threshold hyper-parameter has no consistent effect on the accuracy of the topics. A threshold of 0.5 with both auxiliary features gives the best accuracy for both development sets, therefore for testing, we use these two hyper-parameters.

The test results are displayed in table 4.5. The baseline (Zhang et al., 2022) we compare to uses a logistic regression model where hostnames are features and their values are their degree of stance alignment with the question. The authors use a task specific passage extraction algorithm made specifically for the treatment/condition style topics of the health misinformation track. They retrieve the top 3000 documents with BM25 before extracting their query-dependant passages. They then use a T5 model to calculate the degree to which each document's passage is aligned with the stance of its topic. For each hostname, its top scoring BM25 passage's alignment score will be the respective feature's

Track year		2019			2021		
		Features					
	Threshold	HH	H	N	HH	H	N
Big Bird Aggregation	0.500	**66.6**	54.9	64.7	**82.0**	74.0	74.0
	0.625	62.7	60.8	64.7	78.0	76.0	74.0
	0.750	62.7	66.6	62.7	78.0	78.0	80.0
	0.875	56.9	50.8	62.7	74.0	72.0	80.0

Table 4.4: Accuracy of answer predictions for the questions in the health misinformation track given various hyperparameters. The threshold is the minimum score (passed through a sigmoid function) required for a sentence to be included in the extracted passage. Features are HH meaning both HONCode and Hostnames were used as features, H meaning only hostnames were used and N meaning no auxiliary features were used (So the model only looks at language features). For each track, the other was used as the development set, meaning the model that gave the best performance was saved and used for testing the other track.

Track Year	2019		2021		2022	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Baseline (Zhang et al., 2022)	58.8	60.6	76.0	82.2	**70.0**	**86.4**
Our Approach	66.7	**66.1**	**82.0**	**84.0**	66.0	69.1

Table 4.5: Metrics for our answer prediction pipeline compared to the baseline.

value.

In contrast to the baseline, our proposed approach uses a generalized neural passage extraction approach while also taking language features into account when making an answer prediction. Our proposed approach has better accuracy and AUC metrics than the baseline in the 2019 and 2021 tracks but failed to beat the baseline in the 2022 tracks. It should also be noted that our proposed approach is more computationally expensive than the baseline. Depending on various factors, it would require roughly 3-4 times more computation per query.

4.4 Reranking

In this section, we look at how reranking the documents based on our proposed strategy works in the 2021 and 2022 health misinformation tracks.

In the 2021 track, compared to Pradeep et al. (2021a) our proposed passage extraction mechanism has a better compatibility score (0.073 compared to 0.062). Our helpfulness however ends up being lower.

In the 2021 track, compared to Zhang et al. (2022), the proposed QA pipeline gives a higher compatibility score (0.162 vs 0.129). THis improvement is statistically significant using a two tail paired t-test (pj0..05). The slightly better answer prediction quality coupled with the addition of a neural reranking stage contributed to this uplift in compatibility.

Our α hyperparameter has a fairly big impact on compatibility. For the 2021 track, a

Track Year	2021			2022		
	Help	Harm	Delta	Help	Harm	Delta
Baseline BM25	0.122	0.144	0.022	0.171	**0.140**	0.031
Mono-T5 (Pradeep et al., 2021a)	0.185	0.122	0.062	0.246	0.194	0.052
Big Bird Passage Extration + MT5 Reranking	0.155	0.082	0.073	0.217	0.209	0.007
Trust Reranking (Zhang et al., 2022)	0.198	0.069	0.129	**0.253**	0.177	0.076
BPE+MT5 + QA Reranking, alpha=0.1	0.194	0.061	0.133	0.242	0.153	**0.089**
BPE+MT5 + QA Reranking, alpha=0.2	**0.215**	**0.053**	**0.162**	0.244	0.171	0.073

Table 4.6: compatibility scores for the 2021 and 2022 health misinformation tracks.

larger α boosts documents in agreement with the predicted answer and suppresses those in disagreement. An α of 0.1 boosted helpfulness from 0.155 to 0.194 and reduced harmfulness from 0.082 to 0.61. Increasing α to 0.2 further increase helpfulness to 0.215 and reduced harmfulness to 0.053. In the 2022 track, however, a smaller α was better, giving a delta of 0.089 vs 0.073. This improvement is not statistically significant using a two tail paired t-test (p¿0.05). Looking at the 2022 track, we observe that our Big Bird passage extraction plus Mono-T5 ranking gets very poor helpfulness compared to Mono-T5 passage extraction (0.217 vs 0.246), however with our predicted answer reranking, we boost helpfulness by 0.025 and drop harmfulness by 0.056 which boosts delta compatibility from 0.007 to 0.089, which is a relatively large increase. Given these results, it seems apparent that the QA and reranking modules, as we have implemented them currently, can potentially greatly reduce harmfulness. However, the answer aggregation module in our pipeline still has room for improvement. We leave further improvements for future work.

Chapter 5

Conclusion and Future Work

In this work, we presented a question answering and document retrieval pipeline that uses web documents to answer questions. The two main challenges faced are misinformation and contradicting statements from web documents and the length of web documents. We first proposed a domain filtering approach that helped with reducing misinformation in returned results but did not apply to all varieties of questions.

For our pipeline, we proposed a new query biased passage extraction method that takes advantage of modern more memory efficient transformers that can handle long documents. This approach classifies individual sentences as relevant to the question or not using the context of the entire document. Our architecture has better metrics than the previous best multi-span query-dependant passage extraction neural architecture.

We also propose a new answer prediction and document re-ranking component that while more computationally complex than the previous best approach does yield better metrics.

One of the main directions for improvement is exploring alternatives to BigBird. While effective other sparse attention transformers are less computationally expensive, which can help in making a more practical system.

Another avenue for improvement would be an exploration of auxiliary features. Our approach of pre-pending text to text spans is likely not the best approach for combining manual features and transformer models. Other features such as PageRank, credibility

metrics, etc. should also be investigated to see if they too yield better accuracy. Improving the probability of credible documents appearing in the top results will ensure any reranking strategies will work much better. Given the importance of the answer prediction module, improvements such as using larger transformer models should also be looked into.

References

M. Abualsaud and M. D. Smucker. Exposure and order effects of misinformation on health search decisions. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Rome*, 2019.

A. Andoni and I. Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 793–801, 2015.

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

D. Chen, A. Fisch, J. Weston, and A. Bordes. 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, 2017.

C. L. A. Clarke, S. Rizvi, M. D. Smucker, M. Maistro, and G. Zuccon. Overview of the trec 2020 health misinformation track. In *TREC*, 2020.

C. L. A. Clarke, M. D. Smucker, and M. Maistro. Overview of the trec 2020 health misinformation track. In *TREC*, 2021a.

C. L. A. Clarke, A. Vtyurina, and M. D. Smucker. Assessing top-preferences. *ACM Transactions on Information Systems (TOIS)*, 39(3):1–21, 2021b.

Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1285. URL https://doi.org/10.18653/v1/p19-1285.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics-HLT*, pages 4171–4186, 2019.

J. Dodge, M. Sap, A. Marasović, W. Agnew, G. Ilharco, D. Groeneveld, M. Mitchell, and M. Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, 2021.

R. Epstein, R. E. Robertson, D. Lazer, and C. Wilson. Suppressing the search engine manipulation effect (seme). *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):1–22, 2017.

S. Fox and M. Duggan. Health online 2013. *Health*, 2013:1–55, 2013.

A. Ghenai, M. D. Smucker, and C. L. A. Clarke. A think-aloud study to understand factors affecting online health search. In *Proceedings of the 2020 conference on human information interaction and retrieval*, pages 273–282, 2020.

T. H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526, 2002.

P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM*

international conference on Information & Knowledge Management, pages 2333–2338, 2013.

G. Izacard and É. Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, 2021.

Z.-H. Jiang, W. Yu, D. Zhou, Y. Chen, J. Feng, and S. Yan. Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33:12837–12848, 2020.

J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, 2020.

O. Khattab and M. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020.

O. Khattab, C. Potts, and M. Zaharia. Relevance-guided supervision for openqa with colbert. *Transactions of the Association for Computational Linguistics*, 9:929–944, 2021.

N. Kitaev, L. Kaiser, and A. Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2019.

S. Laversin, V. Baujard, A. Gaudinat, M.-A. Simonet, and C. Boyer. Improving the transparency of health information found on the internet through the honcode: a comparative study. In *User Centred Networked Health Care*, pages 654–658. IOS Press, 2011.

J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

K. Lee, K. Hoti, J. D. Hughes, L. Emmerton, et al. Dr google and the consumer: a qualitative study exploring the navigational needs and online health information-seeking behaviors of consumers with chronic health conditions. *Journal of medical Internet research*, 16(12):e3706, 2014.

K. Lee, M.-W. Chang, and K. Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, 2019.

M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020a.

P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020b.

X. Liu and W. B. Croft. Passage retrieval based on language models. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 375–382, 2002.

Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, 2021.

T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal. 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, 2018.

T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. MS MARCO: A human generated machine reading comprehension dataset. In T. R. Besold, A. Bordes, A. S. d'Avila Garcez, and G. Wayne, editors, *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016. URL http://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

F. A. Pogacar, A. Ghenai, M. D. Smucker, and C. L. A. Clarke. The positive and negative influence of search results on people's decisions about the efficacy of medical treatments. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 209–216, 2017.

R. Pradeep, X. Ma, R. Nogueira, and J. Lin. Vera: Prediction techniques for reducing harmful misinformation in consumer health search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2066–2070, 2021a.

R. Pradeep, R. Nogueira, and J. Lin. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*, 2021b.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. *NIST special publication*, (500225):109–123, 1995.

M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*, 2017a. URL https://openreview.net/forum?id=HJ0UKP9ge.

M. Seo, J. Lee, T. Kwiatkowski, A. Parikh, A. Farhadi, and H. Hajishirzi. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441, 2019.

M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017b. URL https://openreview.net/forum?id=HJ0UKP9ge.

A. Shrivastava and P. Li. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 2321–2329, 2014.

A. Vakili Tahami, K. Ghajar, and A. Shakery. Distilling knowledge for fast retrieval-based chat-bots. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in information retrieval*, pages 2081–2084, 2020.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

E. M. Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82, 1999.

D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

Z. Wang, P. Ng, X. Ma, R. Nallapati, and B. Xiang. Multi-passage bert: A globally normalized bert model for open-domain question answering. 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 5878–5882, 2019.

R. W. White. Belief dynamics in web search. *Journal of the Association for Information Science and Technology*, 65(11):2165–2178, 2014.

R. W. White and A. Hassan. Content bias in online health search. *ACM Transactions on the Web (TWEB)*, 8(4):1–33, 2014.

R. W. White and E. Horvitz. Belief dynamics and biases in web search. *ACM Transactions on Information Systems (TOIS)*, 33(4):1–46, 2015.

P. Yang, H. Fang, and J. Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 1253–1256, 2017.

W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, 2019a.

Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764, 2019b. URL https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

D. Zhang, A. Vakili Tahami, M. Abualsaud, and M. D. Smucker. Learning trustworthy web sources to derive correct answers and reduce health misinformation in search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2099–2104, 2022.

T. Zhao, X. Lu, and K. Lee. Sparta: Efficient open-domain question answering via sparse transformer matching retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 565–575, 2021.

F. Zhu, W. Lei, C. Wang, J. Zheng, S. Poria, and T.-S. Chua. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*, 2021.

M. Zhu, A. Ahuja, D.-C. Juan, W. Wei, and C. K. Reddy. Question answering with long multiple-span answers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3840–3849, 2020.