Reducing Health Misinformation in Search Results

by

Dake Zhang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

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 Contributions

Chapter 4 describes the work solely conducted by myself under the supervision of Professor Mark D. Smucker, which was published as part of a TREC participant notebook paper (Abualsaud, Chen, Ghajar, Phan Minh, Smucker, Vakili Tahami, and Zhang [4]).

Chapter 5 describes the work conducted by myself under the supervision of Professor Mark D. Smucker, also in collaboration with Amir Vakili Tahami and Dr. Mustafa Abualsaud. This work was accepted by SIGIR 2022 as a peer-reviewed short paper and published by the Association for Computing Machinery (ACM) (Zhang, Vakili Tahami, Abualsaud, and Smucker [44]), where my contribution included all stages of this research project, including research idea formulation, code implementation, experiments, result analysis, and paper writing; Amir Vakili Tahami helped double-check my code and write the paper; Dr. Mustafa Abualsaud helped write the paper; Professor Mark D. Smucker formulated the idea, discussed results and helped write the paper. Part of the abstract, part of the introduction written in Chapter 1, part of the related work written in Chapter 2, and part of Chapter 6 are taken from this paper.

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

As a declaration, this thesis includes work of the following 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.

Dake Zhang, Amir Vakili Tahami, Mustafa Abualsaud, and Mark 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 (SIGIR '22)*, pages 2099–2104, 2022, DOI: 10.1145/3477495.3531812.

Abstract

People commonly search the web for answers to health-related questions. With health information being added to the Internet every day, misinformation proliferates and disseminates wildly. Previous work has shown that if health misinformation exists in search results, people can make incorrect decisions, which may cause negative effects on their lives. To reduce health misinformation in search results, we need to be able to find web documents that contain correct information and promote them to higher positions in search results over documents that contain misinformation. In this thesis, we describe our efforts in reducing health misinformation in search results.

First, we describe our participation in the TREC 2021 Health Misinformation Track, which provides a framework for evaluating ranking approaches to reducing health misinformation in search results. This track uses the Compatibility Difference as the primary evaluation metric, which measures the approach's ability to rank correct and credible documents before incorrect and non-credible documents. In the 2021 track, runs that used the provided correct answers were viewed as manual runs. By making use of the known answers and applying a Stance Detection Model for reranking, our manual method achieved a Compatibility Difference score of 0.176, a dramatic improvement over the BM25 baseline with a score of -0.022.

Second, as an extension of our work above, we present a pipeline to automatically derive correct answers by learning trustworthy web sources and then reduce health misinformation in search engine results. Determining the correct answer has been a difficult hurdle to overcome for participants in the TREC Health Misinformation Track. In the 2021 track, automatic runs were not allowed to use the known answer to a topic's health question. By exploiting an existing set of health questions and corresponding known answers, we show it is possible to learn which web hosts are trustworthy, from which we can predict the correct answers to the 2021 health questions with an accuracy of 76%. Using our predicted answers, we can promote documents that we predict contain this answer and achieve a Compatibility Difference score of 0.129, achieving a three-fold performance increase compared with the previous best automatic method with a score of 0.043.

To wrap up, evaluated on the TREC 2021 Health Misinformation Track, our final pipeline achieves new state-of-the-art performance among automatic runs.

Acknowledgements

I would like to thank everyone who offered generous help for this thesis. The completion of this thesis would not have been possible without the support of my supervisor Mark D. Smucker. Professor Smucker has provided numerous constructive suggestions and unwavering guidance during my research and studies. His extensive knowledge of the field of Information Retrieval made me able to think on the shoulders of giants, and his abundant research experience helped me grow faster towards becoming a qualified researcher. I am also grateful to my thesis readers: Charles L. A. Clarke and Gordon V. Cormack for their invaluable advice.

Over the last two years at the University of Waterloo, it was a great pleasure studying and working with many eminent friends and colleagues. Their help deserves my endless appreciation for making my academic journey smooth and enjoyable.

Most importantly, I would like to express my deepest gratitude to my parents Haixia Wang and Fusheng Zhang, for their paramount roles in my life. They respect and fully support every decision of mine, and make me believe in my abilities whenever I face difficulties. I will continue working hard to live up to their expectations.

I also acknowledge the computational support from Compute Canada and the financial support in part from the Natural Sciences and Engineering Research Council of Canada (RGPIN-2020-04665, RGPAS-2020-00080), in part from Mitacs, and in part from the University of Waterloo.

Dedication

To my family and my future love.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

When searching the web for answers to health questions, people can make incorrect decisions that have a negative effect on their lives if the search results contain misinformation. To reduce health misinformation in search results, we need to be able to detect documents with correct answers and promote them over documents containing misinformation.

The TREC Health Misinformation Track (2019-2021) provides a venue and framework for evaluating the performance of ranking methods in terms of their abilities to reduce health misinformation in search results [3, 9, 10]. Topics in this track are formulated as health questions regarding the effectiveness of treatments for health issues. Based on evidence from credible medical sources, organizers also provide a *stance* field for each topic. This *stance* field is viewed as the correct answer to the health question. Table 1.1 shows one sample topic from 50 topics[1] provided by the TREC 2021 Health Misinformation Track [10], whose correct answer is *unhelpful*. In this thesis, we will train and evaluate our methods on the data from the 2019 track and the 2021 track. More details about the TREC Health Misinformation Track will be covered in Chapter 3.

1.1 Research Motivation

Search engine results can either help or hinder people's ability to correctly answer health-related questions, depending on the information quality of returned results [27]. When a search engine's results are biased toward correct information, people are more likely to

[1] All 50 topics in the 2021 track can be found here: https://trec.nist.gov/act_part/tracks/misinfo/misinfo-2021-topics.xml

Topic ID: 123
Query: toothpaste pimple overnight **Description:** Can I get rid of a pimple overnight by applying toothpaste? **Narrative:** A pimple is a pustule that develops when the pores on the skin get clogged. Toothpaste is a paste used to clean teeth. A very useful document would discuss whether or not to use toothpaste to get rid of a pimple. A useful document would help a user decide if applying toothpaste to a pimple is an effective means to clear a pimple by giving information about recommended treatments for pimples, the effect of toothpastes when applied to skin, or both.
Stance: Unhelpful **Evidence:** https://www.hopkinsallchildrens.org/Patients-Families/Health-Library/HealthDocNew/Does-Putting-Toothpaste-on-a-Pimple-Make-It-Go-Awa?id=0

Table 1.1: One sample topic from the TREC 2021 Health Misinformation Track [10]

make a correct decision, but when biased toward incorrect information, people are more likely to make an incorrect decision than if they had not searched in the first place [27]. As shown by White and Hassan [40], biased search results can come about from the bias in the document collection, to how people formulate their queries, to how the retrieval algorithm functions. Likewise, how people respond to search results can be affected by the prevalence of certain answers in the results [1, 15, 17, 35] as well as their own personal biases [7, 21, 41].

Unfortunately, modern commercial search engines can provide users with misinformation in their search results. For example, we used the health question "Can I get rid of a pimple overnight by applying toothpaste?" from Table 1.1 as the input search query to one of the most popular modern search engines, using the browser in the incognito mode. According to the credible evidence in Table 1.1 by medical professionals, toothpaste is viewed as an unhelpful or even harmful treatment for pimples. However, shown in Figure 1.1, the top three results returned by the search engine were a mix of correct information and misinformation, where the misinformation was placed at the top position and the instructions on how to apply the toothpaste on the pimple were even extracted for clear display. This shows that this search engine does not have a proper mechanism to identify misinformation and promote correct information over misinformation, at least for the query in our example.

Since the web is filled with both correct information and misinformation created by people either intentionally or unintentionally, this issue will occur in any search engine

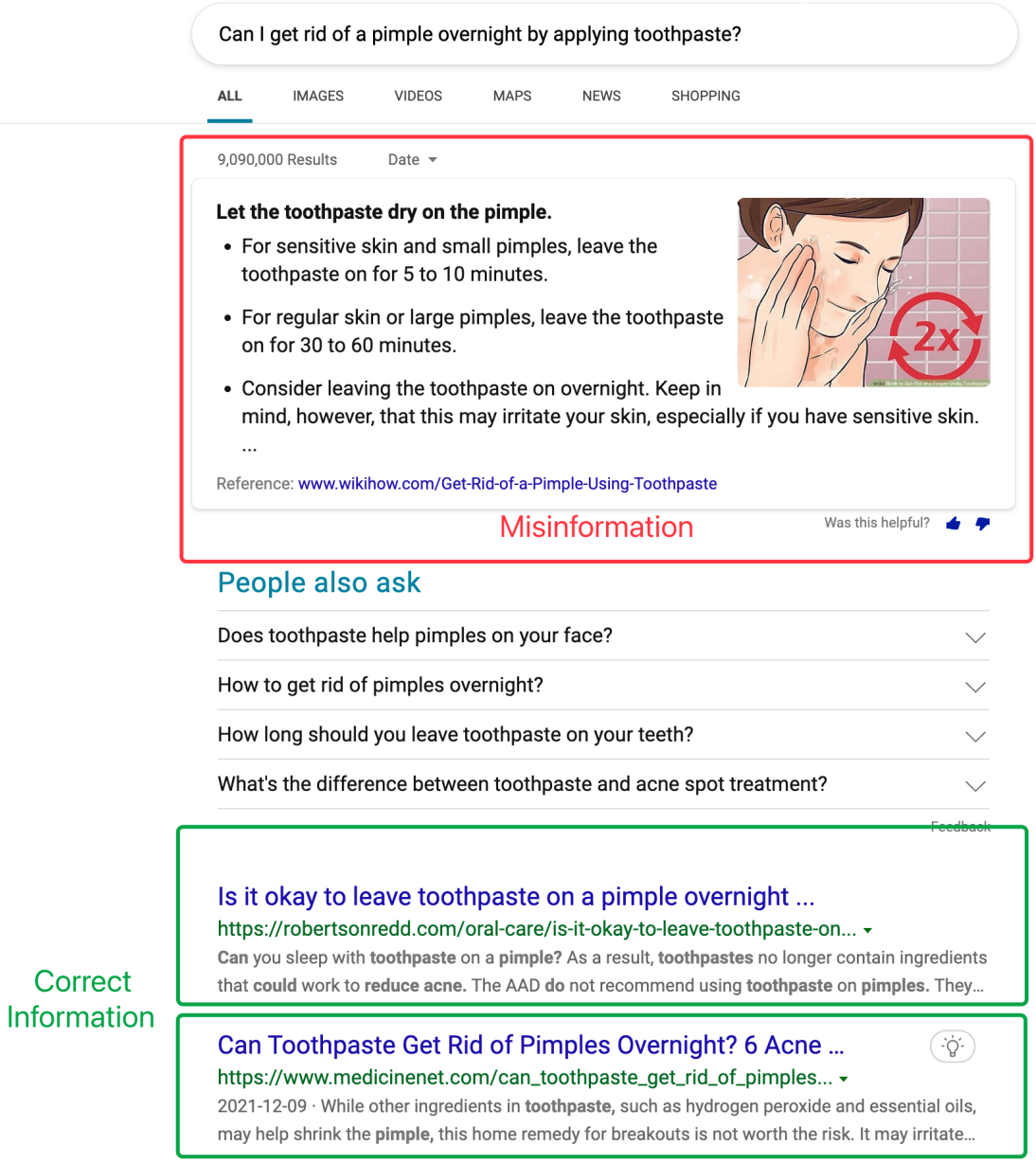

Figure 1.1: Search results from one popular modern search engine (browser in the incognito mode) using the health question in Table 1.1. This screenshot was taken on March 13th, 2022.

BM25 Score	Content
17.211	$\cdots$ *Toothpaste will irritate the skin, and the pimple will probably eventually disappear along with the irritation, but toothpaste is* ***in no way a primary treatment*** *for acne.* $\cdots$ *Toothpaste gives* ***extraordinary results*** *while removing pimples.* $\cdots$
17.205	$\cdots$ *Toothpaste and Baking Soda Baking soda has antiseptic and anti-inflammatory properties that* ***help to clear pimples****.* $\cdots$ *Toothpaste is easily available and an* ***effective remedy to avoid acne****.* $\cdots$
17.065	$\cdots$ *Best Answer: the gel one won't work, you need to plain colgate white toothpaste.* $\cdots$ ***NO! NEVER PUT TOOTHPASTE ON A ZIT!*** $\cdots$ *However, acnes scars can be* ***easily gotten rid of*** *with these 10 simple ways like toothpaste and honey.* $\cdots$
16.938	$\cdots$ *This toothpaste can also be* ***used for the treatment of pimples and acnes to heal them quickly****. Make sure that you are using the white toothpaste instead of toothpaste kind of gel.* $\cdots$
16.928	$\cdots$ *Using any white toothpaste (not gel) on the Zits is quite a popular home treatment for this problem nowadays. For this, you simply have to apply some toothpaste on the Pimples before bed time and let it remain on the face over night. Wash your face the next morning and you will find* ***a great deal of improvement in your Pimples****. The toothpaste should remain on the Zits for at least half an hour.* $\cdots$
16.844	$\cdots$ *The antimicrobial properties present in toothpaste* ***might help in treating your pimples****.* $\cdots$ *Toothpaste* ***helps dry out existing pimples****.* $\cdots$ *"Toothpaste is more complicated than it used to be and* ***can irritate or over-dry your skin****," warns Dr. Day.*

Table 1.2: Top 6 BM25 search results using the health question in Table 1.1. Sentences indicating answers were manually selected for illustration. Key phrases were shown in bold.

that does not have capable strategies to return correct and credible information, where incorrect or non-credible but relevant information is likely to be returned. As a baseline, we performed a BM25 search (with the default implementation in Pyserini [23]) using the same query "Can I get rid of a pimple overnight by applying toothpaste?" on the C4.en.noclean[1] data collection. The top six returned results are shown in Table 1.2, which were also a blend of correct information and misinformation.

If people make incorrect decisions with regard to their health queries, these decisions may cause serious negative impacts on their lives. Therefore, dedicated efforts should be put in place to make search engines more responsible and reliable. Approaches to reducing the rate at which people make incorrect decisions include changes to the search process [21], alerting users to bias in results [16], providing answers directly [12, 19], and the ranking of search results [32]. In this thesis, we focus on the latter approach, i.e., ranking correct information over incorrect information.

1.2 Thesis Overview

As is mentioned above, a responsible and reliable search engine should take multiple factors into consideration when ranking documents to provide relevant, accurate, and correct information, especially for health-related search. In this thesis, using the TREC Health Misinformation Track as the benchmark, we try to develop ranking methods that can promote correct information and reduce misinformation in search results.

Specifically, this thesis is organized as follows.

In Chapter 2, we review related work, including Health-related Search, Stance Detection, and Claim Verification.

In Chapter 3, we cover details of the TREC Health Misinformation Track, such as topics, qrels, and evaluation metrics.

In Chapter 4, we describe our participation in the TREC 2021 Health Misinformation Track [10], where we submitted a manual run. We fine-tuned a Language Model to detect documents' stances towards the health question in the topic, i.e., the effectiveness of the treatment for the health issue. Then, using the provided known answers, we reranked BM25 search results by combining BM25 retrieval scores and predicted stances into a final score, which promoted documents whose stances aligned well with the known answers. This manual run was evaluated by the track organizers and proved to be a strong manual

[1] https://huggingface.co/datasets/allenai/c4

method, which shows that stances are good indicators of whether a document contains correct information or misinformation. However, to be useful in the real world, we need the search system to figure out what the correct answer is by itself, since answers to real-life health questions are not always available and easily accessible.

To address the drawback of our manual method above, in Chapter 5, we present an automatic pipeline to promote correct information over misinformation in search results. By utilizing a set of health questions with known answers, we trained a Trust Model to learn which web hosts are trustworthy. If a hostname has many relevant documents whose stances align well with known answers, then we tend to believe that hostname is trustworthy. When predicting the answer to an unseen health question, we should pay more attention to stances of documents from those trustworthy sources. In our experiment, our Trust Model could predict the correct answers to the 2021 health questions with an accuracy of 76%. Using our predicted answers, we then followed a similar procedure as what we did in Chapter 4 to promote documents that we predicted contained this answer, achieving a Compatibility Difference (the primary evaluation metric for the TREC 2021 Health Misinformation Track) score of 0.129, which is a three-fold increase in performance over the previous best automatic method.

In Chapter 6, we conclude the thesis and point to future directions.

1.3 Contributions

We make the following contributions in this thesis:

- During our participation in the TREC 2021 Health Misinformation Track, our manual method proves that stances are good indicators to distinguish correct information and misinformation. (Chapter 4)
- Besides utilizing the provided known answers, our manual method does not involve additional manual efforts and can be a strong manual run, even though there are manual runs with better performance but they require manual query rewriting. (Chapter 4)
- Evaluated on the 50 health questions from the TREC 2021 Health Misinformation Track, we show it's possible to train a Trust Model to learn which web hosts are trustworthy and then use that knowledge to predict the answers to those health questions based on the web collection. (Chapter 5)

- With the Stance Detection Model and the Trust Model, we present a fully automatic pipeline to promote correct information over misinformation in search results, which achieved new state-of-the-art automatic performance in the TREC 2021 Health Misinformation Track. (Chapter 5)

Chapter 2

Related Work

This chapter covers related work for this thesis, which includes Health-related Search, previous work on Stance Detection (Section 2.2) which our work in Chapter 4 builds upon, and previous work on Claim Verification (Section 2.3) which our work in Chapter 5 builds upon.

2.1 Health-related Search

When people search the web for medical treatments, search engines may return incorrect or misleading information. Previous work has shown that people are prone to make incorrect decisions when the search results are biased towards misinformation [27, 17, 35]. Azzopardi [7] summarized some cognitive biases that users may experience for health-related search, such as Anchoring Bias, Availability Bias, Confirmation Bias, and so on. Ghenai et al. [17] found that when most of the search results express the same opinion, search users tend to interpret this opinion as what most people believe. It will be troublesome if the search engine returns a lot of misinformation but does not retrieve much correct information or put correct information way below the misinformation. Moreover, Abualsaud and Smucker [1] found that reducing users' exposure to misinformation in search results can mitigate their inclination to make incorrect decisions.

To reduce the rate at which people make incorrect decisions for health-related search, researchers have developed many systems for different stages of users' decision-making processes. Epstein et al. [16] found that warning search users of the ranking bias would help them make well-considered decisions. Hashavit et al. [19] developed a machine learning

model to find the correct answer to the health-related query from the medical community. Demner-Fushman et al. [12] developed a question answering system by selecting the best answers from consumer-oriented reliable sources. Besides, Pradeep et al. [32] proposed a multi-stage pipeline for ranking search results to promote correct information, which requires the correct answers to the queries as inputs to the pipeline.

In this thesis, we focus on preventing people from making incorrect health decisions by reducing misinformation in search results, i.e., ranking correct information over incorrect information in the ranked list.

2.2 Stance Detection

The task of automatically detecting attitude or stances expressed in natural language is often an essential component in solution systems for research problems such as Sentiment Analysis [25] and Argument Mining [22]. The definition of Stance Detection varies a lot in different scenarios. Here, we follow Küçük and Can [20]'s definition of the Stance Detection Task: "automatic classification of the stance of the producer of a piece of text, towards a target, into one of these three classes: {*Favor, Against, Neither*}". The key point is to find the stance towards the given target, without being disturbed by stances towards other targets in the same text, which may require inference to a certain extent.

The history of the evolution of Stance Detection Models can be dated back to the era of feature-based learning. Aldayel and Magdy [5] leveraged various online features of social media users to train an SVM model to detect stances in tweets. Before long, neural networks start to dominate this task for their superior capabilities to capture semantic information. Zhang et al. [43] enhanced the BiLSTM stance classifier by integrating a graph convolutional network to learn semantic connections within the tweet. With the fast development of giant pre-trained Language Models in the field of NLP, transformer-based Language Models have become the most popular and effective solutions for developing Stance Detection Models. Allaway and McKeown [6] have shown that simply feeding the contextual conditional encoding of the text and the topic from the BERT model [13] to a feed-forward neural network can achieve better performance than many specially designed neural networks. They further developed the Topic-Grouped Attention (TGA) Net by adding a topic-grouped attention to the BERT model, improving its performance in zero-shot and few-shot settings.

Stance Detection has been used in manual runs to rerank documents since the TREC 2020 Health Misinformation Track [9], achieving superior performance at promoting correct

information while reducing misinformation in search results. When a ranking system is given the correct answer to the health question in the search topic, Pradeep et al. [32] have shown that by manually rewriting the search query to incorporate the correct answer, the T5 [33] model can be fine-tuned to find documents whose stances align with the reformulated query and those documents will be promoted to higher positions in the final ranked list. Their manual run achieved the best manual performance in both the 2020 track and the 2021 track. Unfortunately, their method requires manual rewriting of queries and lacks a way to automatically determine the correct answer, which limits the approach in real-life scenarios. In this thesis, we will provide a solution to those limits.

2.3 Claim Verification

The rate of misinformation being added to the Internet is far beyond the control of traditional human fact-checkers and authorities. At the same time, the dissemination of misinformation has been faster than ever with the broad use of the Internet including social medias, search engines, and so on. Consequently, automated fact-checking systems are in increasingly urgent need to help Internet users and professionals in the combat against misinformation. Automatic Claim Verification (or Fact Verification) has been a heated line of research in recent years [8], which is in close relationship with applications such as Fake News Detection, Rumor Detection, and so forth. The aim of the Claim Verification task is to automatically determine the veracity of a claim using a knowledge base.

A lot of benchmark datasets have been released to foster this line of research, including general factoid claims such as FEVER [36] and RumourEval 2019 [18], news claims such as LIAR [39] and CREDBANK [24], health-related claims such as SCIFACT [37] and COVID-Fact [34], and so on. Many of those benchmark datasets come with controlled evidence sources [39, 37]. The task is typically defined as finding relevant evidence passages, detecting their stances towards the claim (supporting or refuting the claim), and then making a final prediction of the veracity of the claim. For instance, Thorne et al. [36] constructed the FEVER dataset by manually extracting claims from the Wikipedia dataset (June 2017 dump), where the task is to predict the veracity of those claims based on documents from the Wikipedia dataset. Their baseline system consists of three stages: document retrieval (finding the k most relevant documents), evidence sentence selection (selecting the l most similar sentences to the claim from those documents), and textual entailment recognition (predicting the veracity of the claim by recognizing textual entailment between the claim and selected evidence sentences). As an another example, Saakyan et al. [34] constructed the COVID-Fact dataset by collecting 4,086 real-world claims from posts

in the *r/COVID19* subreddit which were accompanied by credible evidence documents, where the task is to find evidence sentences from the provided evidence sets and then predict the veracity of the claims. Similarly, there are two major components in their baseline system: evidence retrieval (retrieving potential source documents and then selecting the most similar sentences to the claim) and veracity prediction (binary classification based on the concatenation of evidence sentences).

As we can see from the baseline systems provided by those benchmark datasets and some successful end-to-end solution systems [31, 38], Stance Detection has been an important and effective component in the pipeline for verifying claims, whose aim is to find the stance alignment of a given text (referred to as evidence) with respect to a claim (target). For instance, as the best submission on the SCIFACT leaderboard[1], the MULTIVERS [38] system predicts the final fact-checking label (either "supports" , "refutes", or "not enough information") through a classification attention head over the joint representation of the claim and the most relevant evidence. Particularly, some benchmark datasets even have sub-tasks that focus on predicting the veracity of the claim given manually selected evidence sentences (gold evidence) [34].

As is mentioned in Section 2.2, in the Health Misinformation Track, some strong manual methods lack the ability to automatically derive the correct answers to the given health questions. In fact, determining those correct answers can be framed as a Claim Verification task, where the claim is that the proposed treatment is helpful for the health issue. However, unlike those mentioned benchmark datasets for automatic claim verification, to our best knowledge, there is no existing curated knowledge base that contains all the credible information we need to determine the correct answers to the health questions in the Health Misinformation Track. Thus, we have to turn to utilize open-domain sources like the web. But we need to be careful when dealing with documents on the Internet, for web documents are prone to include misinformation.

To avoid being affected by the misinformation on the web, researchers have tried different approaches to extract correct and credible information from the web for verifying claims. For example, to utilize relevant evidence articles from the web to verify a claim, Popat et al. [28, 29] jointly modeled several factors for each evidence article, including its stance and language style and the reliability of its source. They defined a formula to compute the reliability of each web source, which was derived from the number of times a web source supported a true claim or refuted a false claim. Similarly, Dong et al. [14] first automatically extracted numerous facts from websites and then modeled the trustworthiness and accuracy of sources using an iterative process, based on the assumption that trustwor-

[1] https://leaderboard.allenai.org/scifact/submissions/public

thy sources contain accurate facts and accurate facts come from trustworthy sources. In Chapter 5, inspired by their research, we show a simplified yet effective method to predict the correct answers to health questions in the TREC 2021 Health Misinformation Track using a web collection. We apply a Machine Learning model to learn the trustworthiness of sources by checking how often their stances align with the correct answers using an existing set of health questions with known answers. We then use the model to predict the answers to new health questions, with stances from trustworthy sources being assigned with greater weights. And based on the predicted answers, we can then rank documents according to their predicted relevance and stance alignment with the correct answers.

Chapter 3

Materials and Methods

Throughout this thesis, all experiments were performed mainly on the data from the TREC 2019 Decision Track [3] (was renamed to the Health Misinformation Track in 2020) and the TREC 2021 Health Misinformation Track [10].

We skip the TREC 2020 Health Misinformation Track [9] because of its divergence in terms of topics and data collections. In response to the misinformation related to COVID-19, the 2020 track focuses on treatments for COVID-19 or SARS-CoV-2, where CommonCrawl News[1] (from January 1st to April 30th, 2020) is used as the data collection. On the one hand, the efficacy for some topics is still controversial in academia and quickly changing with ongoing studies. On the other hand, different from the data collections used in the 2019 track and the 2021 track, the news collection only covers a small portion of documents on the Internet and a short time range (four months). Besides, we used the data collection C4.en.noclean[2] for our experiments because it contains a large enough number of diversified web documents (more than 1 billion). But topics in the 2020 track don't have relevant documents in C4.en.noclean because it was extracted from the April 2019 snapshot[3] of the Common Crawl corpus. Therefore, we performed our experiments on the data from the 2019 track and the 2021 track, skipping the 2020 track.

For topics from the Health Misinformation Track, each is a pair of a health issue and a related treatment (e.g., "aloe vera wounds" from the 2019 track [3]). For each topic, organizers also provide a correct answer that represents the medical consensus on the true efficacy of the treatment according to trusted medical sources (e.g., cochrane.org).

[1] https://commoncrawl.org/2016/10/news-dataset-available/
[2] https://huggingface.co/datasets/allenai/c4
[3] https://commoncrawl.org/2019/04/april-2019-crawl-archive-now-available/

	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 3.1: Statistics of judged documents in qrels. When counting the number of words in a document, we used the plain text of it and removed punctuation and numbers.

Table 3.1 shows some statistics of judged documents in the 2019 qrels and the 2021 qrels respectively.

3.1 TREC 2019 Health Misinformation Track

Previously named as the Decision Track, the 2019 track provides 51 topics[1] for participants. Table 3.2 shows one sample topic from the 2019 track. These topics have three types of efficacy labels: helpful, unhelpful, and inconclusive. There are 17 topics for each efficacy type. In this thesis, we only focus on helpful topics and unhelpful topics.

There are 22,859 judgments in qrels[2], where documents are judged in terms of three aspects: relevance, effectiveness, and credibility. The relevance judgment refers to the document's relevancy to the query, including "highly relevant", "relevant", and "not relevant". The effectiveness judgment refers to the document's answer to the health question, which includes "effective", "inconclusive", "ineffective", and "no information". The credibility judgment refers to the perceived credibility of the document, including "credible" and "not credible". Table 3.1 shows more details on the distributions of some judgments. For consistency, we map "highly relevant" judgments into "very useful" judgments, "relevant"

[1]Available at https://trec.nist.gov/data/misinfo/2019topics.xml
[2]Available at https://trec.nist.gov/data/misinfo/2019qrels_raw.txt

Topic ID: 8
Query: melatonin jet lag **Description:** Can melatonin be used to reduce jet lag? **Narrative:** Jet lag is a fatigue and sleep disorder caused by air travel across several time zones. It has been suggested that melatonin can be used to reduce or prevent the effects of jet lag. Relevant documents should discuss whether taking melatonin can be effective for treating jet lag.
Efficacy: Helpful **Cochrane DOI:** 10.1002/14651858.CD001520

Table 3.2: One sample topic from the TREC 2019 Health Misinformation Track [3]

judgments into "useful" judgments, "effective" judgments into "supportive" judgments, and "ineffective" judgments into "dissuasive" judgments.

The 2019 track uses ClueWeb12-B13[1] as the web collection. Collected in 2012, this data collection includes around 50 million pages, which is a subset of ClueWeb12-Full.

3.2 TREC 2021 Health Misinformation Track

Different from the 2019 track, the 2021 track provides 50 topics[2] with efficacy labels of either helpful or unhelpful. There are 25 topics for each efficacy type. So for consistency of our experiments, we didn't use "inconclusive" topics from the 2019 track.

There are 12,778 judgments in qrels[3], where documents are judged in terms of three aspects: usefulness, supportiveness, and credibility. The usefulness judgment refers to the extent to which the document contains useful information for search users to answer the health question, including "very useful", "useful", and "not useful". The supportiveness judgment refers to whether the document supports or dissuades the use of the treatment for the health issue, which includes "supportive", "neutral", and "dissuades". The credibility judgment refers to the perceived credibility of the document, including "excellent", "good", and "low". Table 3.1 shows more details on the distributions of some judgments. For consistency, we map "dissuades" judgments into "dissuasive" judgments, "excellent"

[1] https://lemurproject.org/clueweb12/

[2] Available at https://trec.nist.gov/data/misinfo/misinfo-2021-topics.xml

[3] Available at https://trec.nist.gov/data/misinfo/misinfo-resources-2021.tar.gz

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, no 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 3.3: Preference ordering of documents from the 2021 track [10]

judgments into "very credible" judgments, and "good" judgments into "credible" judgments.

Due to the budget constraint of NIST, of the track's 50 topics, NIST assessors only judged 35 topics, among which there were 3 topics that did not have any documents considered harmful. As the track's organizers did [10], we only used the remaining 32 topics for evaluation purposes in this thesis.

The 2021 track uses C4.en.noclean[1] as the web collection. Released by Raffel et al. [33] in 2019, this collection has over 1 billion documents extracted from the April 2019 snapshot of the Common Crawl corpus. In this thesis, all BM25 search was performed on this collection.

3.3 Compatibility

Starting from the 2020 track, Compatibility [11] has become the primary evaluation metric to measure each method's ability of promoting correct information over misinformation.

[1] https://huggingface.co/datasets/allenai/c4

Compatibility computes the similarity of a given ranking R to the ideal ranking I using Rank Biased Overlap (RBO) shown below.

$$\text{RBO}(R, I) = (1 - p) \sum_{i=1}^{\infty} p^{i-1} \frac{|I_{1:i} \cap R_{1:i}|}{i}$$

where $I_{1:i}$ denotes the set of top i items in the ideal ranking and $R_{1:i}$ denotes the set of top i items in the given ranking. The *agreement* between two rankings at depth i is defined as the overlap (size of the intersection between $I_{1:i}$ and $R_{1:i}$) divided by i. A standard TREC run has 1000 ranked documents for each topic, so the depth goes down to 1,000 in this case. Thus, RBO can be viewed as a weighted average of the *agreement* across depths from 1 to 1000. The parameter $p \in (0, 1)$ represents searcher patience or persistence. The larger this p is, the more willing the searcher is to look at more results down the ranked list. The default value of p is 0.95, which is roughly equivalent to NDCG@20.

In the 2021 track, the ideal ranking (preference ordering) is constructed by sorting the qrels based on NIST's judgments of usefulness, correctness, and credibility of documents [10], shown in Table 3.3. For example, a document with a score of 10 should be ranked in a higher position than a document with a score of 9.

To compute Compatibility scores, in the 2021 track, organizers created two sets of preference qrels: helpful preference qrels and harmful preference qrels. Helpful preference qrels are taken from judged documents with preference scores greater than zero. Meanwhile, for harmful preference qrels, they used the absolute values of preference scores of judged documents whose preference scores are less than 0. So the harmful preference ordering represents the worst case of ranking, where very useful, incorrect, and very credible documents are placed in top positions in the ranked list. Compatibility (helpful) scores and Compatibility (harmful) scores are then computed based on helpful and harmful preference orderings respectively. In other words, Compatibility (helpful) measures the method's ability of ranking helpful documents in high positions, while Compatibility (harmful) measures the method's ability of ranking harmful documents in high positions. Thus, a good method should have a high Compatibility (helpful) score and a low Compatibility (harmful) score. The Compatibility Difference score is the difference between the Compatibility (helpful) score and the Compatibility (harmful) score, which measures each method's ability to rank correct and credible documents over incorrect and non-credible documents. The greater this difference is, the better the method is at promoting correct and credible information over misinformation.

Chapter 4

Stance Detection and Reranking

In this chapter, we will describe our participation in the TREC 2021 Health Misinformation Track. In Chapter 3, we have introduced the TREC Health Misinformation Track, whose aim is to promote correct information over misinformation among search engine results in the health domain. In the 2021 track, organizers provided 50 topics, each consisting of a health issue and a possible treatment. Each topic also came with a *correct stance*, which was derived from professional and credible medical sources, such as the Cochrane Library[1]. In our participation in the 2021 track, we developed a Stance Detection Model capable of making binary predictions of the stances expressed in web documents with regard to the effectiveness of the treatment to the health issue in the topic. By utilizing the *correct stance*, we reranked documents in the BM25 results based on their stance alignment with the *correct stance* and achieved significant improvement over the BM25 baseline. At the end of this chapter, we will present our post-TREC analysis of this manual method using qrels from the 2021 track.

4.1 Methods

Inspired by the success of the Vera system [32] during the TREC 2020 Health Misinformation Track, we choose to fine-tune the pre-trained T5 language model [33] to detect stances expressed in documents. We formulate it as a binary classification task: given the health topic and a relevant document, the model aims to detect the document's stance towards the treatment for the health issue, i.e., whether or not the document supports the use of

[1] https://www.cochranelibrary.com/

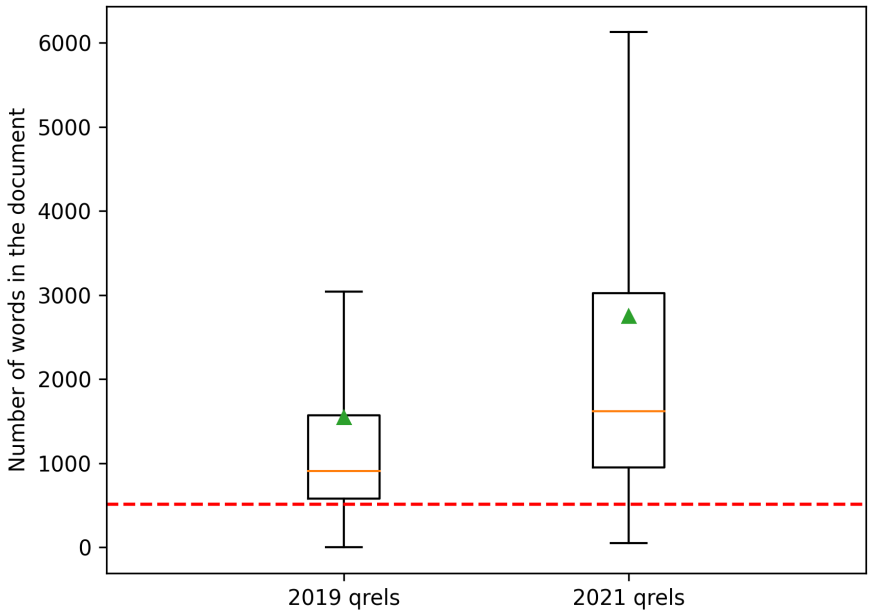

Figure 4.1: Word count boxplots of useful documents in the 2019 qrels and the 2021 qrels. The two green triangles represent the average values. The two short horizontal lines in orange represent the median values. The horizontal dashed line in red represents the input limit of 512 tokens.

the treatment. In this section, we will describe how to build the Stance Detection Model and use it to rerank documents retrieved by BM25 to promote correct information.

4.1.1 Sentence Selection

Most transformer-based Language Models nowadays have an input limit of 512 tokens. However, for web documents used in the Health Misinformation Track, Table 3.1 has shown that the average numbers of words in judged documents exceed greatly the limit of 512 tokens in both years. To have a clearer picture of this issue, we plot two boxplots of the numbers of words in useful documents (judged as "relevant" or "highly relevant" in the 2019 track, judged as "useful" or "very useful" in the 2021 track), shown in Figure 4.1. We can observe that more than 75% of the useful documents in both years contain excessive

numbers of words than the input limit.

One common practice to address this issue is to make a summary that contains the most useful information for the task and has fewer words than the input limit. Therefore, before feeding documents into the Stance Detection Model, we design a heuristic approach for selecting relevant sentences from documents. The intuition is that we need to select sentences that are most relevant to the topic and are most helpful for determining the document's stance towards the use of the treatment for the health issue. Our analysis in Section 4.3.3 confirms the effectiveness of our sentence selection method in improving the Stance Detection Model's classification performance.

A typical useful document usually starts off by introducing the health issue and possible treatments, and then gives its recommendation of promising treatments. To construct a good input to our Stance Detection Model, we need the input to be focused on the part where the stance is clearly expressed and therefore can be easily detected. Thus, we define a list of indicator words consisting of the topic's query tokens and a set of pre-selected stance-related words[1]. Our ablation analysis in Section 4.3.4 shows that those stance-related words help make our sentence selection method work better.

Algorithm 1 shows our heuristic approach in pseudocode and the Python implementation is put in Appendix B. Given a topic (query) and a relevant document, we first perform preprocessing, removing all hyperlinks, numbers, and punctuations, and lowercasing all words. Then we split the document into sentences and score each sentence by counting the total number of occurrences of our indicator words within the sentence. When comparing words in the sentence with our indicator words, we take their stemmed forms using `PorterStemmer`[2] implemented in `nltk.stem.porter`[3]. For example, for the query "*toothpaste pimple overnight*", the sentence "*toothpaste will probably burn and hurt your skin*" will have a score of 2, from "*toothpaste*" and "*hurt*". Finally, we concatenate those top-scoring sentences according to their original orders in the document. If the total number of tokens is still less than 512, we repeatedly add the sentences following the first selected sentence, since those sentences are more likely to be useful for stance detection than sentences before it. Since the TREC Health Misinformation Track only provided document-level judgments, this method is compromising, which can not handle synonyms as neural Language Models can do. Passage-level or sentence-level judgments are needed for fine-tuning a neural Language Model to replace our heuristic method.

[1] Stance-related words: `help, treat, benefit, effective, safe, improve, useful, reliable, evidence, prove, experience, find, conclude, ineffective, harm, hurt, useless, limit, insufficient, dangerous, bad.`

[2] https://tartarus.org/martin/PorterStemmer/

[3] https://www.nltk.org/_modules/nltk/stem/porter.html

Algorithm 1: Sentence Selection

Data:

S ▷ A list of pre-defind stance-related words after stemming
Q ▷ The query field: a list of query terms after stemming
T ▷ Document content after preprocessing: a list of sentences
m ▷ The total number of words in selected sentences
n ▷ The minimum number of words in a selected sentence

Result:

R ▷ Input to the Stance Detection Model: a list of selected sentences

$C(t) \leftarrow 0$; ▷ Tracks the number of words in a sentence
$V(t) \leftarrow 0$; ▷ Tracks the score of each sentence
$R \leftarrow \{\}$; ▷ Input to the Stance Detection Model: a list of selected sentences
for $t \in T$ **do**
    ▷ Loop through each sentence in the document, counting and scoring
    $W \leftarrow$ `nltk.word_tokenize(`t`)`; ▷ Split each sentence into words
    $C(t) \leftarrow$ number of elements in W;
    for $w \in W$ **do**
        $w \leftarrow$ `porter_stemmer.stem(`w`)`; ▷ Stem each word before comparison
        if $w \in S \vee w \in T$ **then**
            $V(t) \leftarrow V(t) + 1$; ▷ Get one point if w is stance-related or in the query
Sort $V(t)$ in an descending order and obtain a corresponding list of indices I;
$x \leftarrow 0$; ▷ Counts the total number of words in selected sentences
for $i \in I$ **do**
    ▷ First round of selection: select sentences based on their scores
    if $V(i) \leq 0$ **then**
        end loop; ▷ Has looped through all sentences with scores greater than 0
    if $x > m$ **then**
        end loop; ▷ Has selected enough sentences
    if $C(i) \geq n$ **then**
        $R \leftarrow R \cup i$; ▷ Only select sentences with more than $n - 1$ words
        $x \leftarrow x + C(i)$; ▷ Update the total number of words in seleted sentences

```
if x < m then
    ▷ Select more sentences if the total number of words is still below the limit
    for t ∈ T do
        ▷ Loop through all sentences according to their original order again
        if t ∈ R then
            continue;                                   ▷ Skip already selected sentences
        if x > m then
            end loop;                                   ▷ Has selected enough sentences
        if C(t) ≥ n then
            R ← R ∪ t;                  ▷ Only select sentences with more than n − 1 words
            x ← x + C(t);   ▷ Update the total number of words in seleted sentences
    Sort R according to the order of sentences in T;
    Concatenate selected sentences in R;
```

4.1.2 Stance Detection Model

The T5 Language Model [33] reframes multiple Natural Language Processing (NLP) tasks into a unified text-to-text framework where different NLP tasks can be prompted. We adapt a method similar to Nogueira et al. [26] in constructing our input to prompt T5 for the Stance Detection task. Our input to the model is framed as:

`stance detection target :` {`query`} `document :` {`selected sentences`}

where {`query`} is the topic's query and {`selected sentences`} is the concatenation of high-scoring sentences described earlier. To obtain binary classification scores from text outputs of T5, we use an approach similar to Pradeep et al. [32]. Specifically, we apply a softmax function on the logits of the word "favor" and the word "against" found in T5's first generated token. This allows our model to output a `supportive_score` and a `dissuasive_score` that sum to 1. These two scores indicate the extent to which the document supports or dissuades the use of the treatment for the health issue. This is a simplified version of the stance detection task defined in Section 2.2. A finer classification (such as three classes: "supportive", "dissuasive", and "neutral") can be implemented in the future work.

We use stance judgments from 2019 qrels for fine-tuning. However, the qrels for those 34 topics (17 helpful topics and 17 unhelpful topics), are heavily imbalanced, with a total

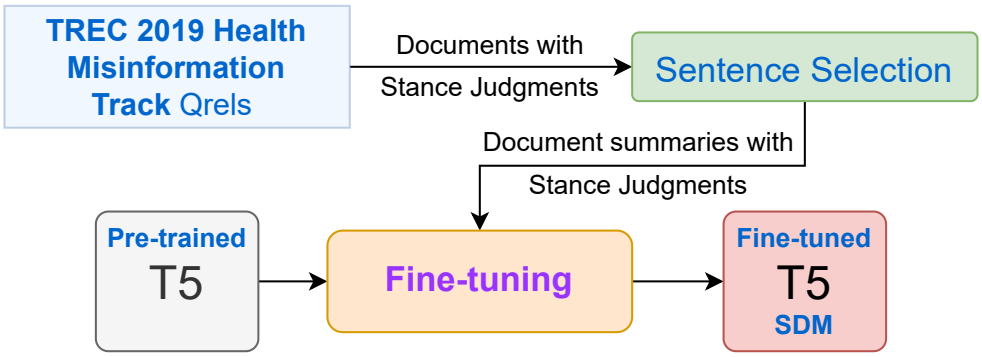

Figure 4.2: Procedure of building the Stance Detection Model, which is fine-tuning the pre-trained T5 language model [33] on sampled qrels from the TREC 2019 Health Misinformation Track.

of 2,078 supportive documents and only 144 dissuasive documents. To prevent the model from biasing towards the majority and improve the model's generalizability, we randomly sample an equal number of supportive and dissuasive documents for each topic and remove document judgments of 5 topics that only have supportive or only have dissuasive documents. Our analysis in Section 4.3.5 supports the importance and necessity of this sampling step. The overall procedure of fine-tuning is shown in Figure 4.2.

4.1.3 Reranking

This is the last step to generate a run. The intuition is that misinformation usually carries different stances from the "truth". For example, documents that support the use of toothpastes to treat pimples are misinformation and should be ranked as low as possible in the search results. Thus, given the correct stance towards a topic and a list of relevant documents with detected stances, we can rerank those documents by promoting documents whose stances are aligned with the correct stance and suppressing documents whose stances are against the correct stance.

Figure 4.3 gives an overview of the pipeline for detecting stances and reranking search results. Given a new topic (the query field in specific), we first retrieve relevant documents using BM25 and then apply our Stance Detection Model on the top 3,000 documents to obtain their stances towards the topic. Finally, we rerank search results based on predicted stance scores, the correct stance, and BM25 scores. Specifically, if the correct stance is "helpful", then documents with greater `supportive_score` will be reranked in

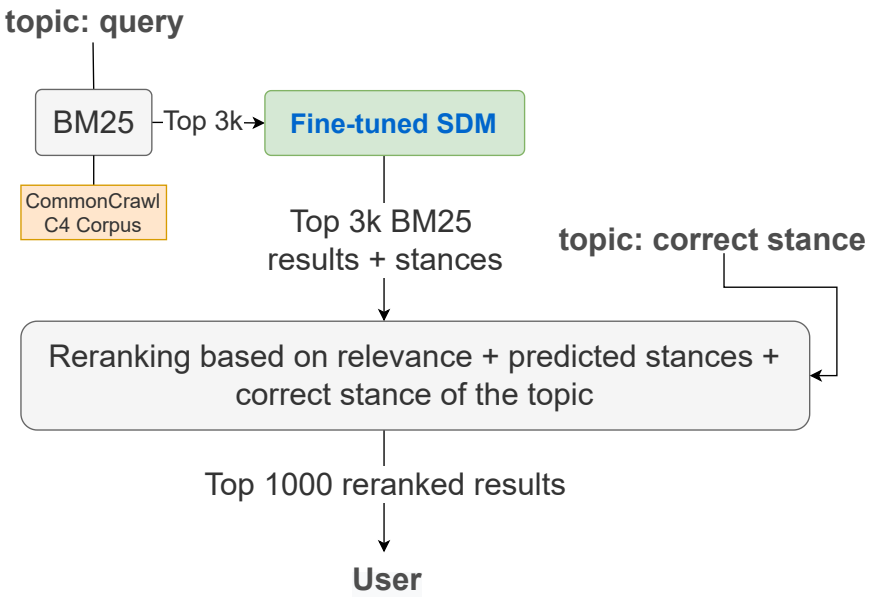

Figure 4.3: Pipeline for detecting stances and reranking search results based on correctness and relevance.

higher positions. Similarly, if the the correct stance is "unhelpful", then documents with greater `dissuasive_score` will be reranked in higher positions. Following this logic, we use the formula below to compute for each document a `correct_score` that reflects the extent to which this document provides the correct information, where we assign the value of `correct_stance` to be 1 if the topic's correct stance is "helpful" and 0 if the topic's correct stance is "unhelpful".

$$\begin{aligned}\texttt{correct\_score} \;=\; &\texttt{supportive\_score} \times \texttt{correct\_stance} + \\ &\texttt{dissuasive\_score} \times (1 - \texttt{correct\_stance})\end{aligned}$$

We then combine `correct_score` with the BM25 score to rerank search results using the formula below:

$$\texttt{final\_score} = \texttt{BM25\_score} \times e^{\texttt{correct\_score} - 0.5}$$

4.2 Experiment

In our experiment, we evaluated the effectiveness of our method on the data from the TREC 2019 and 2021 Health Misinformation Track [3, 10], which have been introduced in Chapter 3. We used the evaluation metric set by the track's organizers, i.e., the Compatibility measure [11].

4.2.1 Data

The 2019 track has 54 topics, and the 2021 track has 50 topics. Each topic is comprised of one health issue and a corresponding treatment (e.g., "toothpaste pimple overnight" in Table 1.1). Each topic also comes with a correct stance (answer) which describes the true efficacy of the treatment based on evidences from credible and trusted medical sources (e.g., https://www.cochrane.org/). Answers to the health topics in the 2019 track are categorized into "helpful", "unhelpful", and "inconclusive", while answers in the 2021 track are either "helpful" or "unhelpful". For consistency, we only used "helpful" or "unhelpful" topics in our experiment. In total, we have 17 "helpful" topics and 17 "unhelpful" topics from the 2019 track, and 25 "helpful" topics and 25 "unhelpful" topics from the 2021 track.

For training and testing our Stance Detection Model, we used qrels from the 2019 track and 2021 track, particularly the "effectiveness" judgments in the 2019 track and "supportiveness" judgments in the 2021 track. In the 2019 track, the "effectiveness" judgment refers to the document's answer to the health question, either "effective", "no information", "ineffective", or "inconclusion". In the 2021 track, the "supportiveness" judgment refers to the document's stance towards the use of the treatment to the health issue, either "dissuades", "neutral", or "supportive". For our Stance Detection Model, we only focused on binary classification. Therefore, we used documents judged as either "effective" or "ineffective" from the 2019 track and documents judged as either "supportive" or "dissuades" from the 2021 track. For consistency and clarity, we mapped "dissuades" into "dissuasive", "effective" into "supportive", and "ineffective" into "dissuasive" for the rest of this thesis.

Both versions of the track use web-based document collections: ClueWeb12-B13[1] is used in the 2019 track and C4.en.noclean[2] is used in the 2021 track. We used the corresponding document collection for each track to retrieve documents in our experiment.

[1] https://lemurproject.org/clueweb12/
[2] https://huggingface.co/datasets/allenai/c4

4.2.2 Data Preprocessing

As is mentioned in Section 4.1.2, to prevent our Stance Detection Model from biasing towards the majority class, we randomly sampled qrels to obtain an equal number of supportive and dissuasive documents (judgments). For stance detection, we only utilized the stance judgments (supportive or dissuasive) from qrels. We used ClueWeb12-B13 for retrieving documents judged in 2019 qrels and C4.en.noclean for retrieving documents judged in 2021 qrels. Specifically, for training and testing our model, we had the following four qrels:

1. **2019 qrels**: contains 3,024 supportive documents[1] and 161 dissuasive documents for 34 "helpful"/"unhelpful" topics.
2. **2019 sampled qrels**: contains 138 supportive documents and 138 dissuasive documents for 29 topics sampled from **2019 qrels**. 5 topics were removed because they had only supportive or only dissuasive documents.
3. **2021 qrels**: contains 3,667 supportive documents and 889 dissuasive documents for 35 "helpful"/"unhelpful" topics[2].
4. **2021 sampled qrels**: contains 497 supportive documents and 497 dissuasive documents for 31 topics sampled from **2021 qrels**. 4 topics were removed because they had only supportive or only dissuasive documents.

These two original qrels and two sampled qrels are used throughout this thesis. The code for sampling qrels can be found in Appendix C.1 and Appendix C.2.

4.2.3 Experiment Settings

Overall, we trained and validated our Stance Detection Model on the 2019 track's data, and used the model to do inference on the 2021 track's data. Our test results could be compared fairly with other runs submitted to the 2021 track because we used exactly the same data available to participants during the 2021 track. We also reported results of cross-validation of our Stance Detection Model solely on the 2021 track's data for post-TREC analysis. Specifically, we had the following three experiment settings:

[1] One supportive document judged in 2019 qrels did not exist in ClueWeb12-B13 and therefore was removed.

[2] 15 topics out of the total 50 topics were not judged by NIST due to its budget constraint.

Fold	2019 Topic ID
Fold 0	1, 3, 12, 22, 34, 47, 50
Fold 1	6, 27, 36, 38, 39, 40, 51
Fold 2	11, 17, 19, 20, 32, 42, 49
Fold 3	13, 18, 28, 29, 41, 44, 45
Fold 4	5, 7, 8, 16, 33, 37

Table 4.1: Stratified random 5-fold cross-validation split of topics in the 2019 track. Inconclusive topics were excluded. Code can be found in Appendix A.1.

Fold	2021 Topic ID
Fold 0	103, 110, 114, 118, 120, 123, 126, 127, 128, 131
Fold 1	101, 108, 112, 113, 115, 129, 133, 139, 143, 150
Fold 2	105, 106, 107, 109, 117, 125, 140, 141, 147, 149
Fold 3	102, 116, 122, 124, 132, 135, 138, 142, 144, 148
Fold 4	104, 111, 119, 121, 130, 134, 136, 137, 145, 146

Table 4.2: Stratified random 5-fold cross-validation split of topics in the 2021 track. Code can be found in Appendix A.2.

(a) **2019 cross-validation** : We randomly split 34 "helpful"/"unhelpful" topics into 5 stratified folds (shown in Table 4.1) and performed cross-validation to find the best set of hyperparameters for our Stance Detection Model.

(b) **2021 test** : Using the Stance Detection Model with hyperparameters obtained above, we evaluated our model and our method (stance detection + reranking) on the 2021 track's data. For retrieving relevant documents in C4.en.noclean, we used the default BM25 implemented in Pyserini [23].

(c) **2021 cross-validation** : We randomly split 50 "helpful"/"unhelpful" topics into 5 stratified folds (shown in Table 4.2) and performed cross-validation to evaluate our approach to build the Stance Detection Model.

Topic splits in Table 4.1 and Table 4.2 were used in all experiments throughout this thesis for fair comparison of different approaches.

4.2.4 Model Hyperparameters

We trained and tested our Stance Detection Model in a zero-shot setting, meaning the data splitting was by topics so that the model couldn't see test topics in its training set. We fine-tuned `T5-Large`[1] using the AdamW optimizer with a learning rate of 2e-5 and a batch size of 16, with Early Stopping based on the F1-macro on the validation set (random 10% of the training set) with a patience of 5.

4.2.5 Evaluation

We evaluated our Stance Detection Model using classifier metrics: True Positive Rate, False Negative Rate, Accuracy, and Area Under the Curve. We evaluated our final run using the Compatibility measure [11] set by the track's organizers: Compatibility (helpful), Compatibility (harmful), and Compatibility Difference (helpful - harmful). We have introduced how those Compatibility scores are computed in Section 3.3. Note that of the 50 topics in the 2021 track, NIST only judged 35 topics, among which there are 3 topics that do not have any documents considered harmful. As the track's organizers did [10], we only considered the remaining 32 topics for evaluation purposes.

4.3 Results & Discussion

In this section, we report our experiment results and analysis of our method. We first report our Stance Detection Model's classification performance, and then compare our runs against other runs submitted to the TREC 2021 Health Misinformation Track. Finally, we present our analysis of some components (e.g., sentence selection, sampling, and so on) used in our method. The success of our method in this chapter lays the foundation of another success in Chapter 4.

4.3.1 Stance Detection

Table 4.3 shows the evaluation results under the three experiment settings mentioned in Section 4.2.3. We can observe that our Stance Detection Model is able to detect the document's stance (either supportive or dissuasive) with good classification performance

[1] https://huggingface.co/t5-large

Setting	Data	TPR	FPR	Accuracy	AUC
2019 cross-validation	2019 sampled qrels	0.827	0.283	0.772	0.839
2021 test	2021 sampled qrels	0.821	0.221	0.800	0.881
	2021 qrels	0.897	0.182	0.882	0.930
2021 cross-validation	2021 sampled qrels	0.746	0.138	0.804	0.895

Table 4.3: Classification performance of the Stance Detection Model under three settings. TPR: True Positive Rate, FPR: False Positive Rate, AUC: Area Under the Curve.

	Run Name	Comp.(helpful)	Comp.(harmful)	Comp.(helpful - harmful)
Auto	`Best-Automatic`	0.195*	0.153*	**0.043***
	`WatSAE-BM25`	0.164	0.123	0.040
	`BM25-Baseline`	0.122	0.144	-0.022
Manual	`Best-Manual`	**0.297***	**0.038***	**0.259***
	`WatSMC-Correct`	0.264†	0.055$^{\diamond}$	0.208†
	`Stance-Reranking`	0.223†	0.047†	0.176†

Table 4.4: Overall performance using the Compatibility metric. Two-tailed paired (per topic) student t-test was performed for statistical analysis. † indicates significant difference from `BM25-Baseline` ($p < 0.01$). $^{\diamond}$ indicates significant difference from `BM25-Baseline` ($p < 0.05$). * indicates the value is from Clarke et al. [10]. **Bold font** indicates the best automatic/manual performance.

(high TPR, low FPR, high accuracy, and high AUC) under 5-fold cross-validation of both tracks. For **2021 test**, we trained our model on the 2019 sampled qrels and evaluated it on the 2021 qrels. We can find that the model trained on 2019 qrels transferred well to the 2021 track (2021 qrels), proving the model's good generalizability. Since the 2021 qrels are heavily imbalanced, those evaluation metrics may not be able to clearly distinguish the performance of different models. Thus, we also evaluated our model on the 2021 sampled qrels, and the results will be used for comparison in the ablation studies later in this section.

4.3.2 Overall Performance

We applied the Stance Detection Model (fine-tuned under Setting b) on the top 3,000 documents retrieved by BM25 and then reranked those documents based on their correctness and relevance. We obtained our run by keeping the top 1,000 documents after reranking, and named it as `Stance-Reranking`. We kept the top 1,000 documents retrieved by BM25 as the baseline, named as `BM25-Baseline`. Table 4.4 shows the Compatibility results of our runs and some good runs submitted to the TREC 2021 Health Misinformation Track:

- `Best-Automatic` (automatic): A point-wise reranking model (monoT5-3B) was fine-tuned on MS MARCO[1] and then on Med-MARCO[2] for relevance ranking, which was used to rerank BM25 results.
- `WatSAE-BM25` (automatic): This is one of our runs submitted to the 2021 track [4]. We created a filter collection which only included documents from domains with an HONcode certification[3] or 13 handpicked health related websites (e.g. `kidshealth.org`). We further expanded this collection by adding relevant documents and removed non-medical documents with a medical text classifier. We ran the default BM25 implemented in Anserini [42] on this filtered collection.
- `Best-Manual` (manual): A label prediction model was trained to predict a label ("true", "weak", or "false") for each document, which was then used to rerank the results from `Best-Automatic`. Note that when monoT5-3B was used to rerank documents, the queries were manually reformulated to include the correct stances.
- `WatSMC-Correct` (manual): This is one of our runs submitted to the 2021 track [4]. We manually assessed documents in terms of usefulness and correctness using our high-recall retrieval system's [2] "Search" and "Discovery" components.

We performed statistical analysis of our runs in the table, since other runs are not publicly available from TREC. Specifically, we performed two-tailed paired (per topic) student t-test of each run against `BM25-Baseline`. Our run `Stance-Reranking` shows that by using our method of stance detection and reranking, we can significantly promote correct information and suppress misinformation in BM25 search results, as is evidenced by the significant improvement of Compatibility scores over `BM25-Baseline`. We also performed the two-tailed paired student t-test between `WatSMC-Correct` and `Stance-Reranking`, and those

[1] https://microsoft.github.io/msmarco/
[2] https://github.com/Georgetown-IR-Lab/covid-neural-ir
[3] https://www.hon.ch/en/

Setting	Data	TPR	FPR	Accuracy	AUC
2019 CV	2019 sampled qrels	0.600 (−27.4%)	0.376 (+32.9%)	0.612 (−20.7%)	0.651 (−22.4%)
2021 Test	2021 sampled qrels	0.801 (−2.4%)	0.523 (+136.7%)	0.639 (−20.1%)	0.729 (−17.3%)
	2021 qrels	0.893 (−0.4%)	0.467 (+156.6%)	0.822 (−6.8%)	0.821 (−11.7%)
2021 CV	2021 sampled qrels	0.608 (−18.5%)	0.265 (+92.0%)	0.671 (−16.5%)	0.761 (−15.0%)

Table 4.5: Classification performance of the Stance Detection Model (without sentence selection) under three settings. TPR: True Positive Rate, FPR: False Positive Rate, AUC: Area Under the Curve. The percentage in brackets represents the relative change compared with the one reported in Table 4.3.

two runs appear to be not significantly different. Additionally, we can see there is a clear performance gap between the best manual run `Best-Manual` and our `Stance-Reranking`. We assume this gap comes from the extra neural reranking stage of `Best-Manual`, where the authors also reformulated the query to incorporate the correct stance to help rerank documents with similar stances in higher positions.

4.3.3 Analysis: Sentence Selection

To analyze the importance and necessity of our sentence selection method, we performed this ablation study where we made the first 512 tokens of documents as inputs to our model without the sentence selection stage. Table 4.5 shows that without sentence selection, the model suffers from higher FPR, lower accuracy, and lower AUC under three settings, which confirms the essential value of the sentence selection method in making the Stance Detection Model more powerful.

4.3.4 Analysis: Stance Words in Sentence Selection

As is mentioned in Section 4.1.1, to select useful sentences as inputs to the Stance Detection Model, we scored each sentence based on the occurrences of query terms and stance words. We have explained that those stance words are used to help find sentences that carry stances towards the topic rather than plain explanations of the health issue or the treatment. Here we performed an ablation study where we only counted the occurrences of query terms (without stance words) during sentence selection. Table 4.6 shows that without considering stance words in scoring sentences, the model experiences mildly higher FPR, lower accuracy, and lower AUC under three settings, which shows that using stance-related

Setting	Data	TPR	FPR	Accuracy	AUC
2019 CV	2019 sampled qrels	0.667 (−19.3%)	0.310 (+10.6%)	0.679 (−12.0%)	0.771 (−8.1%)
2021 Test	2021 sampled qrels	0.833 (−1.5%)	0.272 (+23.1%)	0.781 (−2.4%)	0.875 (−0.7%)
	2021 qrels	0.909 (+1.3%)	0.235 (+29.1%)	0.881 (−0.1%)	0.924 (−0.6%)
2021 CV	2021 sampled qrels	0.755 (+1.2%)	0.192 (+39.1%)	0.782 (−2.7%)	0.874 (−2.3%)

Table 4.6: Classification performance of the Stance Detection Model (without stance words during sentence selection) under three settings. TPR: True Positive Rate, FPR: False Positive Rate, AUC: Area Under the Curve. The percentage in brackets represents the relative change compared with the one reported in Table 4.3.

Setting	Data	TPR	FPR	Accuracy	AUC
2019 CV	2019 sampled qrels	0.984 (+19.0%)	0.779 (+175.3%)	0.603 (−21.9%)	0.827 (−1.4%)
2021 Test	2021 sampled qrels	0.968 (+17.9%)	0.535 (+142.1%)	0.716 (−10.5%)	0.852 (−3.1%)
	2021 qrels	0.985 (+9.8%)	0.516 (+183.5%)	0.887 (+0.6%)	0.892 (−4.1%)
2021 CV	2021 sampled qrels	0.858 (+15.0%)	0.261 (+89.1%)	0.798 (−0.7%)	0.881 (−1.6%)

Table 4.7: Classification performance of the Stance Detection Model (without training data sampling) under three settings. TPR: True Positive Rate, FPR: False Positive Rate, AUC: Area Under the Curve. The percentage in brackets represents the relative change compared with the one reported in Table 4.3.

words to focus the input on sentences with clear stances can improve the classification performance of our Stance Detection Model.

4.3.5 Analysis: Sampling

In Section 4.1.2, we have mentioned that to prevent the model from biasing towards the majority class ("supportive" in this case), we sampled an equal number of supportive documents and dissuasive documents to be the training data. To justify our incentive, we performed the following ablation study. For cross-validation (**2019 cross-validation** and **2021 cross-validation**), we used the original qrels for training topics (four folds) and evaluated on the sampled qrels for test topics (the remaining fold). For **2021 test**, we used the original 2019 qrels for training the model and evaluated the model on 2021 sampled qrels and 2021 qrels separately. Table 4.7 shows that without sampling the training data, the model becomes biased towards the supportive class, as is evidenced by the remarkably

higher TPR and FPR, and therefore has poorer classification performance, as is shown by the lower accuracy and AUC. Note that the higher accuracy on 2021 qrels in **2021 test** does not represent that model is better because 80.5% of the documents in 2021 qrels are supportive documents and the accuracy will increase if predicting more documents to be supportive. That's the reason why we also evaluated the model on the 2021 sampled qrels to have a fair view of the model's classification performance. Thus, those results prove the sampling is necessary for preventing biases of the model.

4.3.6 Analysis: Different Reranking Formulas

In Section 4.1.3, we have presented our formula used to combine `correct_score` and `BM25_score` into a final score. During our participation of the 2021 track, we chose that formula based on the intuition that this formula was commonly used as a way to combine scores and therefore would work well in our case. As part of our post-TREC analysis, to understand how different reranking formulas affect the final Compatibility scores, we experimented with several other reranking formulas:

- `Expo-10`: $\texttt{final\_score} = \texttt{BM25\_score} \times 10^{\texttt{correct\_score}-0.5}$.
- `Stance-Only`: $\texttt{final\_score} = \texttt{correct\_score}$.
- `Plus`: $\texttt{final\_score} = \texttt{BM25\_score} + \texttt{correct\_score}$.
- `W-Plus`: $\texttt{final\_score} = \texttt{BM25\_score} + 100 \times \texttt{correct\_score}$. This is a weighted combination of `BM25_score` and `correct_score`.
- `Scaled-Plus`: $\texttt{final\_score} = \texttt{scaled\_BM25\_score} + \texttt{correct\_score}$. `scaled_BM25_score` is obtained by rescaling BM25 scores of top 3,000 documents into $[0, 1]$, through subtracting each score by the minimum score and then being divided by the range (maximum score - minimum score).
- `Scaled-W-Plus`: $\texttt{final\_score} = \texttt{scaled\_BM25\_score} + 100 \times \texttt{correct\_score}$. This is a weighted combination of `scaled_BM25_score` and `correct_score`.

The Compatibility scores of different runs are shown in Table 4.8, where we also performed two-tailed paired (per topic) student t-test for each run against the `BM25-Baseline` and the `Stance-Reranking` respectively. From the results, we can see that all those reranking formulas can significantly improve the quality of search results. `Stance-Only` shows that even if we don't consider relevance, we can still effectively suppress misinformation (a

Run Name	Comp.(helpful)	Comp.(harmful)	Comp.(helpful - harmful)
BM25-Baseline	0.122	0.144	-0.022
Stance-Reranking	**0.223**†	0.047†	0.176†
Expo-10	**0.223**†	0.042†	0.181†
Stance-Only	0.143$^{\diamond}$	**0.010**$^{\dagger\diamond}$	0.133†
Plus	0.188$^{\dagger\diamond}$	0.109$^{\dagger\diamond}$	0.079$^{\dagger\diamond}$
W-Plus	0.221†	0.027†	0.194†
Scaled-Plus	0.210$^{\dagger\diamond}$	0.055$^{\dagger\diamond}$	0.155$^{\dagger\diamond}$
Scaled-W-Plus	0.213†	0.013†	**0.201**†

Table 4.8: Compatibility scores of runs using different reranking formulas. The top two runs are used as baselines for comparison. † indicates significant difference from BM25-Baseline ($p < 0.05$). $^{\diamond}$ indicates significant difference from Stance-Reranking ($p < 0.05$). **Bold font** indicates the best automatic/manual performance.

big drop in Compatibility-harmful compared with BM25-Baseline, and a mild but significant drop in Compatibility-harmful compared with Stance-Reranking) in search results while slightly promoting correct information, by utilizing stances only. This also proves that stances are excellent indicators for detecting health misinformation.

Since those reranking formulas differ a lot, it's natural that the Compatibility (harmful) scores also vary largely. However, there seems to be an upper cap on the Compatibility (helpful) , which indicates that relevance becomes the bottleneck rather than stances. Therefore, if we want to have more correct information in search results, we need to improve the relevance of top-ranked documents by reranking.

Through the comparison among Plus, W-Plus, Scaled-Plus, and Scaled-W-Plus, we can see that by properly adjusting the weights of relevance and stances in the formula, we can further increase Compatibility-helpful and decrease Compatibility-harmful. Note that since BM25 scores vary a lot from single digit to double digit, the scaling of BM25 scores is equivalent to assigning lower weights to relevance. Therefore, those four runs show that stances indeed play an essential role. But it's still necessary to include relevance in the formula. Otherwise, the result run would degrade into Stance-Only. Additionally, instead of heuristically assigning weights to relevance and stances, we can build a learning-to-rank model to learn weights for various signals (relevance, stances, and probably other aspects) to rerank documents.

4.4 Summary

In this chapter, we have presented our participation of the TREC 2021 Health Misinformation Track and post-TREC analysis of several components in our method. By utilizing stances, our method can effectively promote correct information and suppress misinformation in search results, significantly better than the BM25 baseline. Our experiment results and analysis have demonstrated that stances are powerful indicators for detecting misinformation and play a vital role during reranking. However, our method relies on the provided *correct stance* for each topic, but in real-life scenarios, that answer is not always available or easily accessible, which limits the practical value of our method. In the next chapter, to offset this weakness, we will present an automatic pipeline to derive correct answers to reduce misinformation in search results.

Chapter 5

Deriving Correct Answers

In Chapter 4, we have shown that stances are good indicators of whether a web document is correct information or misinformation. Specifically, given the correct answer, we are able to rerank documents based on the extent to which their detected stances align with the correct answer, leading to a strong manual run. However, in practice, those correct answers are not always available or easily accessible. Determining the correct answer has been a difficult hurdle to overcome for participants in the TREC Health Misinformation Track. In the 2021 track, automatic runs were not allowed to use the known answer to a topic's health question. As a result, among submissions to the 2021 track, the top automatic run had a compatibility-difference score of 0.043 while the top manual run, which used the known answer, had a score of 0.259.

In this chapter, we present an automatic pipeline to reduce health misinformation in search results. By using an existing set of health questions and their known answers, we show it is possible to learn which web hosts are trustworthy, from which we can predict the correct answers to the 2021 health questions with an accuracy of 76%. Using our predicted answers, we can promote documents that we predict contain this answer and achieve a compatibility-difference score of 0.129, which is a three-fold increase in performance over the best previous automatic method. This pipeline consists of two major components, a Stance Detection Model (SDM) and a Trust Model (TM). The Stance Detection Model aims to detect a document's stance on the efficacy of the topic, while the Trust Model is used to aggregate predicted stances across different hostnames to learn which hostnames to trust and therefore predict an answer to the topic's health question, whether the treatment is helpful or unhelpful to the health issue.

5.1 Methods

In this section, we will present the details of our automatic pipeline, which is developed based on our methods in Chapter 4. As is mentioned earlier, the pipeline depends mainly on two models: a Stance Detection Model (SDM) and a Trust Model (TM). We use the same Stance Detection Model, based on the T5 Language Model [33], with the best configuration explained in Chapter 4. Same as before, we use the Stance Detection Model to detect a document's stance on the efficacy of the treatment for the health issue. Meanwhile, the Trust Model is designed to learn which hostnames are trustworthy by looking at the predicted stance scores of documents from different hostnames, and then predict an answer of whether or not the treatment is helpful for the health issue.

5.1.1 Trust Model

When we see an unfamiliar question, we usually tend to believe the answers from trustworthy sources. For instance, `www.webmd.com` is known to provide credible and comprehensive medical information written by experienced professionals. Therefore, documents from this hostname are more likely to be correct than documents from other sources, such as personal blogs and shopping websites. But how do we figure out which sources are trustworthy when we don't have an existing list of trustworthy sources? One possible way is to use our existing knowledge. If we have a list of questions with known answers, we can quickly determine whether a document is correct or incorrect. The more correct documents a hostname has, the more trustworthy it will be.

Therefore, we design the Trust Model to automatically learn which hostnames are trustworthy sources that often give correct health information. And then from those trustworthy sources, the model can further predict the answer to the health question. In Chapter 4, our experiments have demonstrated that stances are good indicators of correct information or misinformation. For the input to our Trust Model, we continue to use stances as proxies for determining correctness. To be specific, suppose we have several topics with correct answers for training. For example, we have the topic "toothpaste pimple overnight" with the correct answer "unhelpful". First, for a retrieved relevant document, we use the Stance Detection Model to detect its stance, whether it is supporting or dissuading the use of toothpaste to treat pimples. Then, with the correct answer "unhelpful", we can further infer whether this document is correct or not. In other words, if the document supports the use of the toothpaste to treat pimples, it will be considered as incorrect, and then the trustworthiness of its hostname will be decreased. Otherwise, the document will be viewed

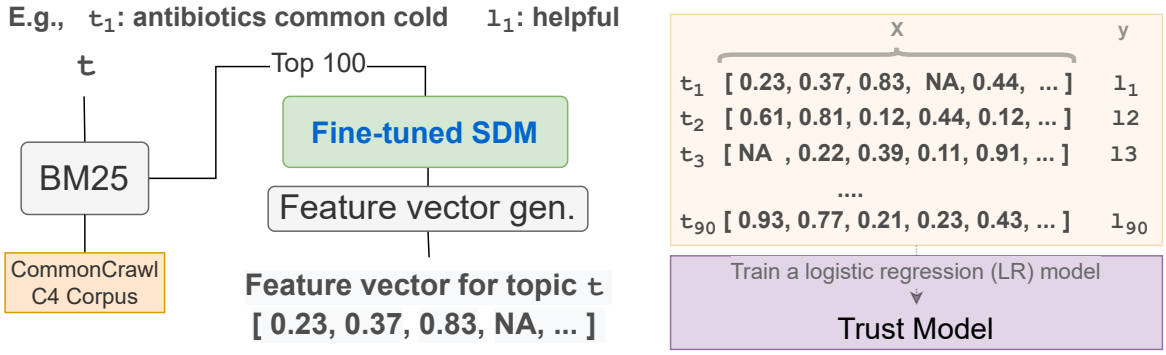

Figure 5.1: Procedure of building the LR-based Trust Model, where the feature vectors are generated from predicted stances of relevant documents from various hostnames.

as correct and its hostname will more likely be trustworthy. Having training topics with known answers and their relevant documents with predicted stances, we are able to train a Machine Learning model to act as the Trust Model.

Figure 5.1 gives an overview of the procedure to build the feature vectors and train the Trust Model, using the Logistic Regression Model as an example. For each training topic, we first retrieve the top 100 documents using the default BM25 implemented in Pyserini [23], along with their URLs. We focus on the top 100 documents for a trade-off between avoiding irrelevant documents and incorporating documents from diversified sources. We further extract the hostnames from their URLs. Then we apply our Stance Detection Model to predict the stances ($\texttt{supportive\_score} \in [0, 1]$) of these 100 documents. Next, these stances are used to construct a feature vector for the topic, where each element corresponds to a hostname and its value is a re-scaled $\texttt{scaled\_supportive\_score}$ of the stance prediction from our Stance Detection Model, calculated by the following formula:

$$\texttt{scaled\_supportive\_score} = 2 \times \texttt{document\_supportive\_score} - 1$$

So the range of this $\texttt{scaled\_supportive\_score}$ is between -1 and 1. If there are multiple documents from the same hostname among those top 100 documents, we choose the top-most document for that hostname (i.e., the most relevant document according to BM25 scores). Thus, the size of the 1-dimensional feature vector is the number of distinct hostnames in the top 100 results across all training topics. The default feature value is 0 if that element's hostname does not appear in the top 100 BM25 results for a specific topic, meaning neutral stance (that's 0.5 before re-scaling). We train the Trust Model using the feature vector as the independent variable and the binary correct answer (1 for helpful and

0 for unhelpful) as the dependent variable. The Trust Model can be based on other Machine Learning models, such as Support Vector Machine, Random Forest, and so on. We'll compare different models' performance in Section 5.3.1. The positive probability output from the Trust Model is denoted as the `helpful_probability` of the topic, indicating the possibility of the topic to be "helpful". The Trust Model is built with the aim that it can learn which hostnames provide more correct health information and update its weights accordingly.

5.1.2 Answer Prediction and Reranking

For predicting the answer to a new health question (new topic), the Trust Model uses the learned weights to aggregate stances from various hostnames to predict the final answer. In essence, the model leverages the wisdom of the crowd, i.e., hostnames, while emphasizing trustworthy sources when making its prediction. The Logistic Regression Model is chosen because of its good performance and interpretability. Having the predicted answer, we then are able to rerank documents based on their stance alignment with the predicted answer and relevance to the topic, similar to what we have done in Section 4.1.3. Finally, we obtain a ranked list of relevant documents where correct information is promoted and misinformation is suppressed.

Figure 5.2 shows the overall pipeline of document retrieval, answer prediction, and reranking when it comes new topics. For a new topic (the query field), we first retrieve the top 3,000 BM25 documents from the web collection, C4.en.noclean in specific for our experiments. Then we apply the Stance Detection Model on those documents to obtain their binary stances towards the health question, i.e., whether the treatment is helpful or unhelpful for the health issue. Next, we take the top 100 most relevant documents as inputs to our Trust Model for predicting the answer to the health question. We extract their hostnames and construct the feature vectors in the same way that we have done for training the Trust Model, while ignoring hostnames that are not seen in the training set. With weights learned from the training data, the Trust Model is able to give a `helpful_probability` of the topic, indicating the possibility of the treatment being "helpful" to the health issue. Finally, we rerank BM25 search results using stance scores from the Stance Detection Model, the predicted answer from the Trust Model, and BM25 scores. To combine those values, for each document, we use the following formula to compute a `correct_score` that reflects our confidence that this document provides the correct information.

$$\begin{aligned} \texttt{correct\_score} \;=\; & \texttt{supportive\_score} \times \texttt{helpful\_probability} + \\ & \texttt{dissuasive\_score} \times (1 - \texttt{helpful\_probability}) \end{aligned}$$

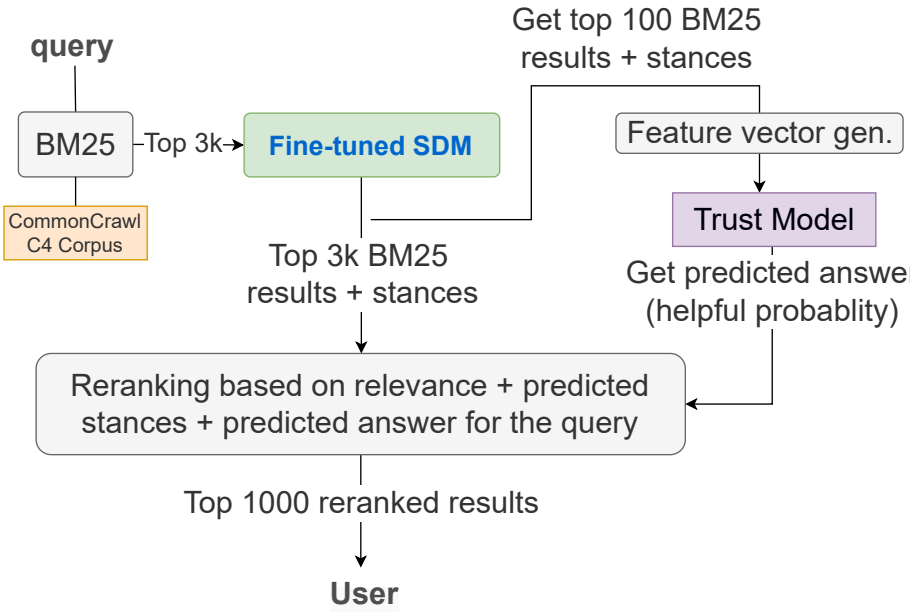

Figure 5.2: Procedure of predicting answers to new queries and reranking search results based on predicted correctness and relevance.

Where `supportive_score` and `dissuasive_score` are from our Stance Detection Model, and `helpful_probability` is from our Trust Model. Note that this formula is a general form of the one we use in Section 4.1.3. That's to say, if we use the provided correct answers, the `helpful_probability` should be 1 for helpful topics and 0 for unhelpful topics. Finally, we combine `correct_score` with `BM25_score` to rerank search results using the same formula in Section 4.1.3:

$$\texttt{final\_score} = \texttt{BM25\_score} \times e^{\texttt{correct\_score} - 0.5}$$

5.2 Experiment

Same as Chapter 4, we evaluated the effectiveness of our pipeline at promoting correct information over misinformation on the data from the TREC 2019 and 2021 Health Misinformation Track [3, 10], where the Compatibility measure [11] was used as the primary evaluation metric. The Stance Detection Model was developed in the same way as in Chapter 4.

5.2.1 Data

We evaluated our pipeline on the same two sets of topics used in Section 4.2.1, 34 topics from the 2019 track and 50 topics from the 2021 track. For training our Stance Detection Model, we followed the same method as in Chapter 4 and used the same data.

For training our Trust Model, to ensure there were enough training examples, we made use of an existing set of health questions with known answers from White and Hassan [40], which were selected from Cochrane Reviews. There are 122 helpful topics, 67 inconclusive topics, and 59 unhelpful topics. Similarly, we focused only on those helpful topics and unhelpful topics. Some of those topics overlap with topics in the 2019 track or in the 2021 track. So we removed overlapping topics that appear in **2019 topics** or **2021 topics**, and sampled a balanced subset, obtaining 45 helpful topics and 45 unhelpful topics for training our Trust Model. In total, for each helpful and unhelpful category, we have 17 topics from the TREC 2019 Health Misinformation Track [3], 25 topics from the TREC 2021 Health Misinformation Track [10], and 45 topics from White and Hassan [40]. For retrieving relevant documents to help predict the answer, we performed BM25 search on C4.en.noclean which is the web collection used in the 2021 track, since this web collection is large enough for retrieving relevant documents.

5.2.2 Experiment Settings

For the training and inference of our Stance Detection Model, we followed the same regime in Section 4.2.3. For training our Trust Model, we used the 90 sampled topics from White and Hassan [40] with relevant documents retrieved from C4.no.clean. The size of the feature vector was 3,847, which was also the number of distinct hostnames in those 9,000 relevant documents (top 100 BM25 results per topic and 90 topics in total). To sum up, we have 90 feature vectors, one for each training topic, and 90 corresponding answer labels, either helpful or unhelpful. We had three experiment settings shown below, compatible with those in Section 4.2.3.

(a) **2019 cross-validation**: We used the same stratified random 5-fold cross-validation split of 34 topics shown in Table 4.1. For each iteration, we trained a Stance Detection Model on qrels of the topics in the 4 training folds. Then we applied that model to predict stances of those 9,000 relevant documents of topics from White and Hassan [40] to construct feature vectors to train our Trust Model. Next, we retrieved the top 100 BM25 documents on C4.en.noclean for each test topic. Similarly, we applied the Stance Detection Model on those documents and built test feature vectors for the

Trust Model. Finally, we used the Trust Model to predict the answers to those test topics. We aggregated the predicted answers of five iterations for evaluation.

(b) **2021 test** : We trained the Stance Detection Model on **2019 sampled qrels** and trained the Trust Model in a similar way as described in **2019 cross-validation**. Next, we retrieved the top 3000 BM25 documents on C4.en.noclean for each **2021 topic**. Similarly, we applied the Stance Detection Model on those documents to obtain their predicted stances. We used the top 100 BM25 documents to build test feature vectors for the Trust Model. After that, we used the Trust Model to predict the answers to **2021 topics**. With BM25 scores, predicted stances, and predicted answers, we further reranked those top 3,000 BM25 documents for each topic to obtain a run for the 2021 track. This run could be used for comparison with other submitted runs during the 2021 track.

(c) **2021 cross-validation**: We followed a similar procedure to the one described in **2019 cross-validation** except that we used the 50 topics in the 2021 track with stratified random 5-fold cross-validation split shown in Table 4.2

5.2.3 Evaluation

We evaluated our Trust Model using standard classifier metrics: True Positive Rate, False Negative Rate, Accuracy, and Area Under the Curve. Same as Section 4.2.5, we evaluated our final run using the Compatibility measure [11]: Compatibility (helpful), Compatibility (harmful), and Compatibility Difference (helpful - harmful). As is mentioned before, in the 2021 track, NIST only judged 35 topics out of 50 topics provided by organizers, among which there are 3 topics that do not have any documents considered harmful. As the track's organizers did [10], we only considered the remaining 32 topics for Compatibility evaluation.

5.3 Results & Discussion

In this section, we'll report our experiment results and post-TREC analysis of our pipeline. We first compare the classification performance of Trust Models based on different Machine Learning algorithms. Then we use the Trust Model based on Logistic Regression to generate an automatic run and compare it with other good runs submitted to the TREC 2021 Health Misinformation Track. Finally, we perform analysis of two components of the Trust Model: top k relevant documents for constructing the feature vector and scaling of stance scores.

Setting	Model	TPR	FPR	Accuracy	AUC
2019 cross-validation	`Logistic Regression`	**0.412**	0.235	0.588	0.612
	`Support Vector Machine`	0.353	0.118	**0.618**	**0.657**
	`Random Forest`	0.176	**0.059**	0.559	0.567
	`Multi-layer Perceptron`	**0.412**	0.294	0.559	0.585
2021 test	`Logistic Regression`	**0.640**	0.120	**0.760**	0.822
	`Support Vector Machine`	0.480	**0.040**	0.720	0.825
	`Random Forest`	0.480	**0.040**	0.720	0.802
	`Multi-layer Perceptron`	0.560	**0.040**	**0.760**	**0.838**
2021 cross-validation	`Logistic Regression`	**0.800**	0.520	0.640	**0.757**
	`Support Vector Machine`	**0.800**	0.400	**0.700**	0.691
	`Random Forest`	0.640	**0.240**	**0.700**	0.697
	`Multi-layer Perceptron`	0.680	0.320	0.680	0.741

Table 5.1: Classification performance of Trust Models based on different Machine Learning algorithms under three settings. TPR: True Positive Rate, FPR: False Positive Rate, AUC: Area Under the Curve. **Bold font** indicates the best performance under each setting.

5.3.1 Answer Prediction

We have tried the following four popular and powerful Machine Learning algorithms (implemented in `scikit-learn`[1]) to build the Trust Model:

- `Logistic Regression`: default configuration (`tol=1e-4, solver='lbfgs', max_iter=100`) except that `penalty='none'`.
- `Support Vector Machine`: default configuration (`C=1.0, tol=1e-3, max_iter=-1`) except that `kernel='linear', probability=True`.
- `Random Forest`: default configuration (`n_estimators=100, criterion='gini'`) except that `random_state=42`.
- `Multi-layer Perceptron`: default configuration (`tol=1e-4, max_iter=200, batch_size='auto'`) except that `solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(10,), random_state=42`.

[1] https://scikit-learn.org

	Hostname	Weight
1	www.cochrane.org	2.7860
2	emedicine.medscape.com	2.3674
3	patient.info	1.9708
4	experts.mcmaster.ca	1.8269
5	www.everydayhealth.com	1.6509
6	www.drugbank.ca	1.5490
7	annals.org	1.2226
8	www.bioportfolio.com	1.1857
9	scholars.duke.edu	1.1856
10	www.tripdatabase.com	1.1263
...	...	...
3838	ccfacra.org	-0.8080
3839	link.springer.com	-0.8412
3840	painmuse.org	-0.8924
3841	www.goldbamboo.com	-0.9205
3842	www.healthystock.net	-0.9254
3843	researchers.uq.edu.au	-0.9276
3844	profiles.ucsf.edu	-0.9960
3845	spotidoc.com	-1.0425
3846	dailymed.nlm.nih.gov	-1.0448
3847	www.scribd.com	-1.0681

Table 5.2: Top 10 and bottom 10 hostnames ranked by weights learned by the LR-based Trust Model trained on White and Hassan [40] Topics

Table 5.1 shows the classification performance of different versions of the Trust Model. We can observe that except `Random Forest`, all other versions are generally good at predicting correct answers of the given health questions (with high TPR, low FPR, high accuracy, and high AUC) across three settings. Meanwhile, the similar performance across `Logistic Regression`, `Support Vector Machine`, and `Multi-layer Perceptron` indicates that the the Logistic Regression algorithm is powerful enough for predicting answers based on the input stances, since SVM and MP are much more powerful but did not yield much better performance in our case. Therefore, we choose the Logistic Regression algorithm to build the Trust Model because of its good performance, simplicity, and interpretability.

To illustrate what the Trust Model has learned, we list the top 10 and bottom 10

	Run Name	Comp.(helpful)	Comp.(harmful)	Comp.(helpful - harmful)
Automatic	Trust-Pipeline	0.198†	**0.069**†	**0.129**†
	Best-Automatic	0.195*	0.153*	0.043*
	WatSAE-BM25	0.164	0.123	0.040
	BM25-Baseline	0.122	0.144	-0.022
Manual	Best-Manual	**0.297***	**0.038***	**0.259***
	WatSMC-Correct	0.264†	0.055$^{\diamond}$	0.208†
	Stance-Reranking	0.223†	0.047†	0.176†

Table 5.3: Overall performance using the Compatibility metric. Two-tailed paired (per topic) student t-test was performed for statistical analysis. † indicates significant difference from BM25-Baseline ($p < 0.01$). $^{\diamond}$ indicates significant difference from BM25-Baseline ($p < 0.05$). * indicates the value is from Clarke et al. [10]. **Bold font** indicates the best automatic/manual performance.

hostnames ranked by weights learned by the LR-based Trust Model when being trained on the White and Hassan [40] Topics in Table 5.2. We observe that those top-ranked hostnames are indeed credible sources, especially www.cochrane.org being assigned with the largest weight. However, in the bottom-ranked hostnames, we can see that the Trust Model can make mistakes. For example, dailymed.nlm.nih.gov is certainly a trustworthy source of information but has a large negative weight. Additionally, it should be noted that for some hostnames, such as profiles.ucsf.edu, documents from individual authors may not be subject to strict editorial processes. So the document quality is not consistently good or bad for such hostnames. Since our Trust Model assigns the same weight for every document from the same hostname, it fails to handle this case properly. Overall, while some host weights appear not to represent credibility perfectly, in the aggregate, the Trust Model is still able to predict correct answers for a majority of health questions, evidenced by the 76% prediction accuracy on the 50 topics from the 2021 track.

5.3.2 Overall Performance

Using our pipeline under Setting b, we obtained an automatic run for the TREC 2021 Health Misinformation Track and named it as Trust-Pipeline. Table 5.3 demonstrates the Compatibility results of our runs (including those from Chapter 4) and some good runs submitted to the 2021 track. We have described those runs in Section 4.3.2.

We performed two-tailed paired (per topic) student t-test of each our run against

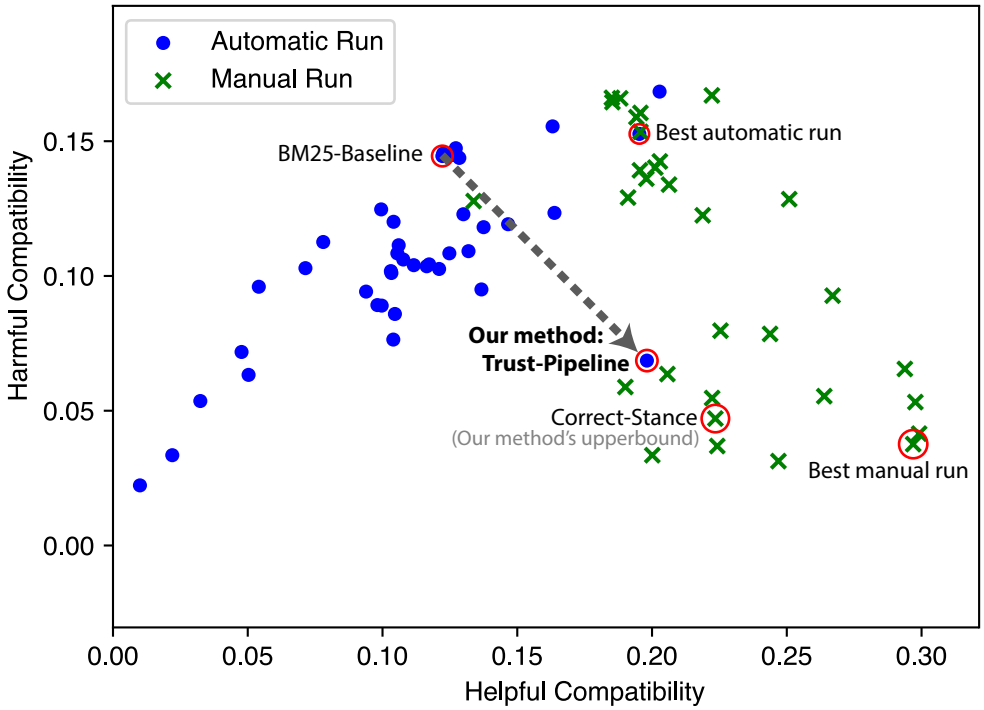

Figure 5.3: Automatic and manual runs submitted to the TREC 2021 Health Misinformation Track.

`BM25-Baseline`. We were unable to perform statistical analysis of other runs due to the fact that per-topic evaluation data of those runs was not publicly available from TREC. Our run `Trust-Pipeline` demonstrates that through our pipeline of stance detection, answer prediction, and reranking, we can effectively suppress misinformation in BM25 search results, as is reflected by the significant drop of the harmful Compatibility score compared with `BM25-Baseline`. Note that `Stance-Reranking` from Chapter 4 is the upper bound of our pipeline in this chapter because `Trust-Pipeline` will have the same performance as `Stance-Reranking` if our Trust Model can predict answers to health questions in the 2021 track with 100% accuracy.

We plotted our runs together with all the runs submitted to the 2021 track in Figure 5.3, using different symbols to separate automatic runs and manual runs. Runs closer to the bottom right corner of the figure generally have higher helpful Compatibility scores and lower harmful Compatibility scores, and therefore are considered better runs. We can see there is a clear gap between automatic runs and good manual runs. However, with our fully automatic pipeline, we are able to rerank the `BM25-Baseline` and move it from the cluster

Setting	Model	TPR	FPR	Accuracy	AUC
2019 cross-validation	$k = 10$	0.118	**0.176**	0.471	0.427
	$k = 50$	0.294	0.235	0.529	0.484
	$k = 100$	**0.412**	0.235	**0.588**	0.612
	$k = 200$	**0.412**	0.294	0.559	**0.633**
	$k = 300$	**0.412**	0.353	0.529	0.630
2021 test	$k = 10$	0.280	**0.120**	0.580	0.566
	$k = 50$	0.600	0.160	0.720	0.788
	$k = 100$	0.640	**0.120**	**0.760**	**0.822**
	$k = 200$	0.680	0.240	0.720	0.794
	$k = 300$	**0.800**	0.280	**0.760**	0.814
2021 cross-validation	$k = 10$	**0.920**	0.800	0.560	0.704
	$k = 50$	0.680	0.600	0.540	0.685
	$k = 100$	0.800	0.520	0.640	0.757
	$k = 200$	0.760	0.560	0.600	0.762
	$k = 300$	0.760	**0.400**	**0.680**	**0.773**

Table 5.4: Classification performance of the LR-based Trust Model using different k values under three settings. TPR: True Positive Rate, FPR: False Positive Rate, AUC: Area Under the Curve. **Bold font** indicates the best performance under each setting.

of automatic runs into the cluster of strong manual runs with low harmful Compatibility scores. With more powerful Trust Models (higher prediction accuracy), `Trust-Pipeline` can be moved closer to its upper bound `Correct-Stance`. In together, Table 5.1 and Figure 5.3 show that our method `Trust-Pipeline` achieves a new high for automatic runs on this task. Our method has a Compatibility (helpful) score comparable to the previous best automatic run, and reduces the Compatibility (harmful) score to levels on par with strong manual runs.

5.3.3 Analysis: Top *k*

When building the feature vector for the Trust Model, we used the stances of top $k = 100$ BM25 documents for each topic. Intuitively, a smaller k will make the model focus on highly relevant documents but limit the number of hostnames seen in the training set. However, a larger k will have the risk of including irrelevant documents, whose stances to the topic should not affect the Trust Model. Table 5.4 shows our experimentation of

Setting	Model	TPR	FPR	Accuracy	AUC
2019 cross-validation	`w/o scaling`	0.176	**0.235**	0.471	0.453
	`w/ scaling`	**0.412**	**0.235**	**0.588**	**0.612**
2021 test	`w/o scaling`	0.440	**0.120**	0.660	0.741
	`w/ scaling`	**0.640**	**0.120**	**0.760**	**0.822**
2021 cross-validation	`w/o scaling`	0.160	**0.080**	0.540	0.696
	`w/ scaling`	**0.800**	0.520	**0.640**	**0.757**

Table 5.5: Classification performance of the LR-based Trust Model with or without scaling under three settings. TPR: True Positive Rate, FPR: False Positive Rate, AUC: Area Under the Curve. **Bold font** indicates the best performance under each setting.

different k values. We can see that across all three settings, the model starts to achieve good classification performance from $k = 100$, but that performance does not further increase when k becomes larger. This confirms our assumption that a small k will limit the number of hostnames in the training data and therefore the model can not utilize many useful hostnames in the test data (those hostnames will have the default value of 0 if not seen in the training data). In addition, the model seems to be quite robust when k becomes greater than 100, where many irrelevant documents will be included when generating feature vectors.

5.3.4 Analysis: Scaling

As is mentioned in Section 5.1.2, we used the formula:

$$\texttt{scaled\_supportive\_score} = 2 \times \texttt{document\_supportive\_score} - 1$$

to change the range of document-level stance scores. Table 5.5 shows the performance of Trust Models with and without such scaling. We observe that this scaling is indeed necessary to make the model work. The reason is that if a hostname does not appear in the top 100 BM25 documents for a specific topic, the default value for that hostname in the feature vector will be 0. Before scaling, a stance score of 0 means the document strongly discourages the use of the treatment for the health issue. But we need the model to ignore hostnames that don't appear in the top 100 BM25 documents for the topic. So after scaling, the default value 0 represents the neutral stance. And 0 in the feature vector can be properly neglected by the Logistic Regression algorithm.

5.4 Summary

In this chapter, our work demonstrates that in the limited domain of the TREC Health Misinformation Track, we can predict answers to unseen questions from the misinformation laden web by learning trustworthy web hosts, and then use these predicted answers to reduce misinformation in search results. Using the top 100 BM25 ranked documents with their predicted stances towards health questions, our Trust Model can predict correct answers with an accuracy of 76% to 50 health questions from the TREC 2021 Health Misinformation Track. With the predicted answers and predicted document stances, we are able to rerank a BM25 baseline and obtain an automatic run that achieves a significant increase in performance over the best previous automatic run.

Chapter 6

Conclusion and Future Work

In this thesis, we have presented our work to promote correct information and reduce misinformation for health-related search, from a manual method to a fully automatic pipeline. In this chapter, we will summarize our work and point to directions for the future work.

6.1 Summary

Our first work, described in Chapter 4, shows the viability of using stances as indicators of whether a document contains misinformation or not. Using a Stance Detection Model, we are able to rerank BM25 search results to effectively increase the ratio of correct information and reduce misinformation in high positions. The only missing puzzle that prevents our manual method to be an automatic method is how to figure out the correct answers to the health questions automatically.

Inspired by the success of using stances as proxies of correct (or incorrect) information, described in Chapter 5, we take the idea of learning trustworthy sources via exogenous signals and utilize it in a simpler form, where we train a Trust Model to learn trustworthy hostnames by seeing if they contain information consistent with a set of health questions with known answers. Using this Trust Model, we can predict the probability of a new topic (health question) to be helpful and rerank documents based on the extent to which the documents' stances align with our prediction. Our method is fully automatic and does not use the provided correct answers to produce a run. We are able to find correct (helpful) information on par with the best automatic run submitted to the TREC 2021 Health Misinformation Track while reducing the amount of incorrect (harmful) results

to a level similar to that of strong manual runs which directly utilized the knowledge of provided correct answers and involved manual rewriting of queries. This performance places our method far above the existing automatic methods and is comparable to some strong manual runs. To our knowledge, our method represents a new state-of-the-art for automatic methods in the TREC 2021 Health Misinformation task.

6.2 Future Work

In this section, we outline some promising directions for future work. We believe our pipeline can be further improved with the following work done in the future.

6.2.1 Neural Search

In our experiments, we used the BM25 score to represent a document's relevancy to the search topic. It's known that bag-of-words retrieval functions like BM25 can not properly handle synonyms and lack the ability to do basic inference in natural language. Many neural search models have demonstrated their remarkable performance on some search engine benchmark datasets, such as MS MARCO[1]. Therefore, we can use a neural search model to replace the BM25 search or rerank BM25 results to improve the relevancy of highly-ranked documents. At the same time, we can also use that neural search model to replace our heuristic sentence selection method to find sentences that carry the stances in each document, which can be used as inputs to our Stance Detection Model.

6.2.2 Stance Detection

In this thesis, we simplify the Stance Detection Model by designing it to make binary prediction (either supportive or dissuasive) of document stances. In fact, relevant documents can have other stances, such as neutral stances or no stance on the use of the treatment. For example, a relevant document may only introduce the health issue without mentioning any of the treatment. Thus, in the future, we can build a more powerful Stance Detection Model by making finer prediction (supportive, dissuasive, neutral, and so on) and also by fine-tuning a pre-trained Language Model more capable than T5.

[1] https://microsoft.github.io/msmarco/

6.2.3 Trust Model

The Trust Model we build in Chapter 5 is a rudimentary attempt of the idea of learning trustworthy hostnames by observing the frequencies of their documents in consistent with a set of health questions with known answers. Many factors are ignored for our Trust Model in this thesis. For instance, our Trust Model does not discriminate different relevancy levels of documents in the top 100 BM25 search results. Intuitively, the stance of a more relevant document should have higher weights during the aggregation process to reach the final prediction. Hence, we can build a better Trust Model by including more input signals, such as relevancy scores and credibility estimates based on the language usage.

6.2.4 Learning to Rank

In Section 4.3.6, we have tried different formulas to combine stance scores and BM25 scores. Using a fixed and heuristically selected formula limits our approach's flexibility in other applications. With the recent studies on learning-to-rank models for ranking tasks, we can build a learning-to-rank model to merge various signals (relevance, stances, and probably other useful signals for detecting correct information or misinformation) to rank documents with the aim to promote correct information and suppress misinformation.

References

[1] Mustafa Abualsaud and Mark D Smucker. Exposure and Order Effects of Misinformation on Health Search Decisions. In *Workshop on Reducing Online Misinformation Exposure*, 2019. 6 pages.

[2] Mustafa Abualsaud, Nimesh Ghelani, Haotian Zhang, Mark D Smucker, Gordon V Cormack, and Maura R Grossman. A System for Efficient High-Recall Retrieval. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1317–1320, 2018.

[3] Mustafa Abualsaud, Christina Lioma, Maria Maistro, Mark D. Smucker, and Guido Zuccon. Overview of the TREC 2019 Decision Track. In *TREC*, 2019. 19 pages.

[4] 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*, 2021. 18 pages.

[5] Abeer Aldayel and Walid Magdy. Your Stance is Exposed! Analysing Possible Factors for Stance Detection on Social Media. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW): 1–20, 2019.

[6] Emily Allaway and Kathleen McKeown. Zero-Shot Stance Detection: A Dataset and Model using Generalized Topic Representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8913–8931, 2020.

[7] Leif Azzopardi. Cognitive Biases in Search: A Review and Reflection of Cognitive Biases in Information Retrieval. In *Proceedings of the 2021 conference on human information interaction and retrieval*, pages 27–37, 2021.

[8] Giannis Bekoulis, Christina Papagiannopoulou, and Nikos Deligiannis. A Review on Fact Extraction and Verification. *ACM Computing Surveys (CSUR)*, 55(1):1–35, 2021.

[9] Charles L.A. Clarke, Saira Rizvi, Mark D. Smucker, Maria Maistro, and Guido Zuccon. Overview of the TREC 2020 Health Misinformation Track. In *TREC*, 2020. 11 pages.

[10] Charles L.A. Clarke, Maria Maistro, and Mark D. Smucker. Overview of the TREC 2021 Health Misinformation Track. In *TREC*, 2021. 12 pages.

[11] Charles L.A. Clarke, Alexandra Vtyurina, and Mark D. Smucker. Assessing Top-k Preferences. *ACM Transactions on Information Systems (TOIS)*, 39(3):1–21, 2021.

[12] Dina Demner-Fushman, Yassine Mrabet, and Asma Ben Abacha. Consumer health information and question answering: helping consumers find answers to their health-related information needs. *Journal of the American Medical Informatics Association*, 27(2):194–201, 2020.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[14] Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang, Wilko Horn, Camillo Lugaresi, Shaohua Sun, and Wei Zhang. Knowledge-Based Trust: Estimating the Trustworthiness of Web Sources. *Proc. VLDB Endow.*, 8(9):938–949, 2015.

[15] Tim Draws, Nava Tintarev, Ujwal Gadiraju, Alessandro Bozzon, and Benjamin Timmermans. This Is Not What We Ordered: Exploring Why Biased Search Result Rankings Affect User Attitudes on Debated Topics. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 295–305, 2021.

[16] Robert Epstein, Ronald E Robertson, David Lazer, and Christo Wilson. Suppressing the Search Engine Manipulation Effect (SEME). *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):1–22, 2017.

[17] Amira Ghenai, Mark D Smucker, and Charles LA 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.

[18] Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. RumourEval 2019: Determining Rumour

Veracity and Support for Rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 845–854, 2019.

[19] Anat Hashavit, Hongning Wang, Raz Lin, Tamar Stern, and Sarit Kraus. Understanding and Mitigating Bias in Online Health Search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 265–274, 2021.

[20] Dilek Küçük and Fazli Can. Stance Detection: A survey. *ACM Computing Surveys (CSUR)*, 53(1):1–37, 2020.

[21] Annie YS Lau and Enrico W Coiera. Do People Experience Cognitive Biases while Searching for Information? *Journal of the American Medical Informatics Association*, 14(5):599–608, 2007.

[22] John Lawrence and Chris Reed. Argument Mining: A Survey. *Computational Linguistics*, 45(4):765–818, 2020.

[23] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362, 2021.

[24] Tanushree Mitra and Eric Gilbert. CREDBANK: A Large-Scale Social Media Corpus With Associated Credibility Annotations. *Proceedings of the International AAAI Conference on Web and Social Media*, 9(1):258–267, 2021.

[25] Saif M Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. Stance and Sentiment in Tweets. *ACM Transactions on Internet Technology (TOIT)*, 17(3):1–23, 2017.

[26] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, 2020.

[27] Frances A. Pogacar, Amira Ghenai, Mark D. Smucker, and Charles 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*, page 209–216, 2017.

[28] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. Where the Truth Lies: Explaining the Credibility of Emerging Claims on the Web and Social Media. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1003–1012, 2017.

[29] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. CredEye: A Credibility Lens for Analyzing and Explaining Misinformation. In *Companion Proceedings of the The Web Conference 2018*, pages 155–158, 2018.

[30] Ronak Pradeep, Xueguang Ma, Xinyu Zhang, Hang Cui, Ruizhou Xu, Rodrigo Nogueira, and Jimmy Lin. H_2oloo at TREC 2020: When all you got is a hammer... Deep Learning, Health Misinformation, and Precision Medicine. In *TREC*, 2020. 11 pages.

[31] Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. Scientific Claim Verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103, 2021.

[32] Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy 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, 2021.

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[34] Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. COVID-Fact: Fact Extraction and Verification of Real-World Claims on COVID-19 Pandemic. 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 2116–2129, 2021.

[35] Qiurong Song and Jiepu Jiang. How misinformation density affects health information search. In *Proceedings of the ACM Web Conference 2022*, pages 2668–2677, 2022.

[36] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a Large-scale Dataset for Fact Extraction and VERification. 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 809–819, 2018.

[37] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, 2020.

[38] David Wadden, Kyle Lo, Lucy Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. MultiVerS: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 61–76, 2022.

[39] William Yang Wang. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, 2017.

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

[41] Ryen W White and Eric Horvitz. Belief Dynamics and Biases in Web Search. *ACM Transactions on Information Systems (TOIS)*, 33(4):1–46, 2015.

[42] Peilin Yang, Hui Fang, and Jimmy 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.

[43] Bowen Zhang, Min Yang, Xutao Li, Yunming Ye, Xiaofei Xu, and Kuai Dai. Enhancing Cross-target Stance Detection with Transferable Semantic-Emotion Knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3188–3197, 2020.

[44] Dake Zhang, Amir Vakili Tahami, Mustafa Abualsaud, and Mark 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*, page 2099–2104, 2022.

APPENDICES

Appendix A

Python Code for Splitting Topics

We used `sklearn.model_selection.StratifiedKFold` from `scikit-learn`[1] to obtain stratified five folds with a random seed of 42, preserving the ratio of helpful topics to unhelpful topics. We also used the `NumPy`[2] library to process the data. The Python code is shown below.

A.1 Code for Splitting 2019 topics

```
import numpy as np
from sklearn.model_selection import StratifiedKFold

topics = [1, 3, 5, 6, 7, 8, 11, 12, 13, 16, 17, 18, 19, 20,
          22, 27, 28, 29, 32, 33, 34, 36, 37, 38, 39, 40,
          41, 42, 44, 45, 47, 49, 50, 51]
topic_answers = [0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1,
                 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0,
                 0, 1, 0, 1, 1, 0]
# 0: unhelpful, 1: helpful
topics = np.array(topics)
topic_answers = np.array(topic_answers)

stratifiedKFold = StratifiedKFold(n_splits=5, random_state=42,
                                  shuffle=True)
for index, (training_topics_idx, test_topics_idx) \
```

[1] https://scikit-learn.org/
[2] https://numpy.org/

```
        in enumerate(stratifiedKFold.split(topics, topic_answers)):
    training_topics = topics[training_topics_idx]
    test_topics = topics[test_topics_idx]
    print(f'Fold [{index}]: {test_topics}')
```

A.2 Code for Splitting 2021 topics

```
import numpy as np
from sklearn.model_selection import StratifiedKFold

topics = [101, 102, 103, 104, 105, 106, 107, 108, 109, 110,
          111, 112, 113, 114, 115, 116, 117, 118, 119, 120,
          121, 122, 123, 124, 125, 126, 127, 128, 129, 130,
          131, 132, 133, 134, 135, 136, 137, 138, 139, 140,
          141, 142, 143, 144, 145, 146, 147, 148, 149, 150]
topic_answers = [0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
                 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,
                 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1,
                 1, 1, 1, 1, 0, 0, 1, 0]
# 0: unhelpful, 1: helpful
topics = np.array(topics)
topic_answers = np.array(topic_answers)

stratifiedKFold = StratifiedKFold(n_splits=5, random_state=42,
                                  shuffle=True)
for index, (training_topics_idx, test_topics_idx) \
        in enumerate(stratifiedKFold.split(topics, topic_answers)):
    training_topics = topics[training_topics_idx]
    test_topics = topics[test_topics_idx]
    print(f'Fold [{index}]: {test_topics}')
```

Appendix B

Python Code for Sentence Selection

We have provided the pseudocode for the Sentence Selection approach to construct summarized inputs to our Stance Detection Model in Algorithm 1. Here, we provide our Python implementation of it.

```
import re
import numpy as np
import nltk
from nltk.stem.porter import PorterStemmer


porter_stemmer = PorterStemmer()
min_sentence_words = 4
# Sentences with fewer than 4 words will be removed.
max_input_length = 512
# The maximum number of words in the generated summary

# A list of stance-related words
stance_words = ['help', 'treat', 'benefit', 'effective', 'safe',
                'improve', 'useful', 'reliable', 'evidence',
                'prove', 'experience', 'find', 'conclude',
                'ineffective', 'harm', 'hurt', 'useless',
                'limit', 'insufficient', 'dangerous', 'bad']
stance_words = [porter_stemmer.stem(word) for word in stance_words]

text = 'sample text'       # Plaintext of the web document
query = 'sample query'     # From the <query> field of the topic
query_terms = [porter_stemmer.stem(query_token.lower())
               for query_token in nltk.word_tokenize(query)]

```

```
26
27 # Start Preprocessing and scoring each sentence
28 text = re.sub(r'https?://\S+|www\.\S+', ' ', text)
29 sentences = []
30 for line in text.split('\n'):
31     for sentence in nltk.sent_tokenize(line):
32         sentence = sentence.lower().strip()
33         sentences.append(sentence)
34 scores = []
35 word_count = []
36
37 processed_sentences = []
38 for sentence in sentences:
39     sentence = re.sub(r'[^A-Za-z]', ' ', sentence)
40     sentence = re.sub(r'\s+', ' ', sentence)
41     sentence = sentence.lower().strip()
42     processed_sentences.append(sentence)
43
44     words = nltk.word_tokenize(sentence)
45     word_count.append(len(words))
46     score = 0
47     for word in words:
48         word = porter_stemmer.stem(word)
49         if word in stance_words or word in query_terms:
50             score += 1
51     scores.append(score)
52
53
54 # Start selecting sentences based on their scores
55 scores = np.array(scores)
56 sorted_indices = np.argsort(scores)[::-1]
57 selected_indices = []
58
59 total_word_count = 0
60 for idx in sorted_indices:
61     if scores[idx] <= 0:
62         break
63     if total_word_count > max_input_length:
64         break
65     if word_count[idx] >= min_sentence_words:
66         selected_indices.append(idx)
67         total_word_count += word_count[idx]
68
69 if total_word_count < max_input_length:
70     idx = min(selected_indices) if len(selected_indices) > 0 else 0
```

```
    while idx < len(scores):
        if idx in selected_indices:
            idx += 1
            continue
        if total_word_count > max_input_length:
            break
        if word_count[idx] >= min_sentence_words:
            selected_indices.append(idx)
            total_word_count += word_count[idx]
        idx += 1

selected_indices.sort()
summary = ' '.join(processed_sentences[index] for index in
   selected_indices)
```

Appendix C

Python Code for Sampling Qrels

We used `pandas.DataFrame.sample` from `pandas`[1] to sample an equal number of supportive and dissuasive judgments from qrels. We also used the `NumPy`[2] library to process the data. The Python code is shown below.

C.1 Code for Sampling 2019 Qrels

```
import numpy as np
import pandas as pd


dataset = pd.read_csv('../data/2019qrels_docs.csv')
dataset = dataset[dataset.stance.isin([1, 3])]
dataset = dataset.dropna()
dataset.stance = dataset.stance.map({1: 0, 3: 1})
dataset = dataset.reset_index(drop=True)

topics = sorted(set(dataset.topic_id.values))

selected_indices = []
for topic in topics:
    part = dataset[dataset.topic_id == topic]
    stance_set = set(part.stance.values)
    if 0 in stance_set and 1 in stance_set:
```

[1] https://pandas.pydata.org/
[2] https://numpy.org/

```
        count = [part[part.stance == i].shape[0]
                 for i in [0, 1]]
        minimum = min(count)
        for i in [0, 1]:
            sample_num = min(minimum,
                             part[part.stance == i].shape[0])
            selected_part = part[part.stance == i].\
                sample(sample_num, replace=False, random_state=42,
                       axis=0)
            selected_indices.extend(selected_part.index.tolist())

selected_indices = np.array(sorted(selected_indices))
dataset = dataset.loc[selected_indices, :]

dataset.to_csv('../data/2019qrels_docs_stance_balanced.csv',
               index=False)
```

C.2 Code for Sampling 2021 Qrels

```
import numpy as np
import pandas as pd


dataset = pd.read_csv('../data/2021qrels_docs.csv')
dataset = dataset[dataset.stance.isin([0, 2])]
dataset = dataset.dropna()
dataset.stance = dataset.stance.map({0: 0, 2: 1})
dataset = dataset.reset_index(drop=True)

topics = sorted(set(dataset.topic_id.values))

selected_indices = []
for topic in topics:
    part = dataset[dataset.topic_id == topic]
    stance_set = set(part.stance.values)
    if 0 in stance_set and 1 in stance_set:
        count = [part[part.stance == i].shape[0]
                 for i in [0, 1]]
        minimum = min(count)
        for i in [0, 1]:
            sample_num = min(minimum,
                             part[part.stance == i].shape[0])
            selected_part = part[part.stance == i].\
                sample(sample_num, replace=False, random_state=42,
```

```
                          axis=0)
            selected_indices.extend(selected_part.index.tolist())

selected_indices = np.array(sorted(selected_indices))
dataset = dataset.loc[selected_indices, :]

dataset.to_csv('../data/2021qrels_docs_stance_balanced.csv',
               index=False)
```