UWaterlooMDS at the TREC 2022 Health Misinformation Track

Amir Vakili Tahami[‡], Dake Zhang[†], and Mark D. Smucker[‡]

Department of Management Sciences[‡] School of Computer Science[†]
University of Waterloo, Ontario, Canada

Abstract

In this paper, we report our participation in the TREC 2022 Health Misinformation Track. With the aim to foster research on retrieval algorithms to promote correct information over misinformation for health-related queries, this year's track had two tasks: web retrieval and answer prediction. We reused our previous method with minor modifications to create our baselines. To overcome some limitations of our previous methods, we investigated a document-aware sentence-level passage extraction model based on the BigBird transformer. The upgraded pipeline with this model achieved our best automatic performance on the web retrieval task but failed to beat our baselines on the answer prediction task. Meanwhile, our manual runs still outperformed our automatic runs by great margins on both tasks, showing room for further improvements.

1 Introduction

TREC Health Misinformation Track [1, 3, 4] aims to foster research on designing and building retrieval systems that are capable of retrieving documents that aid users in reaching the correct decisions for their health questions. Given a question, these systems must identify documents that agree with the current consensus of the medical community. And when displaying results on a search engine results page, they should suppress incorrect documents and promote correct ones to reduce the risk of users reaching harmful decisions regarding their health issues.

Different from previous years, the correct answers to topics this year were not provided by the track organizers. Participants had to determine the correct answers themselves and retrieve the correct documents based on their predicted answers. Specifically, there were two tasks this year: answer prediction and web retrieval. Another difference was that topics this year were no longer framed as "X (potential treatment) for Y (health issue)". They are health-related questions in more general forms, such as "Topic 154: Can cell phones cause cancer?" and "Topic 156: Can mosquito bites make you sick?". This shift made our previous approach [10] no longer directly applicable, where we assumed the topics to be in the format of "X (potential treatment) for Y (health issue)". Therefore, this year, we replaced the heuristic sentence selection component of our stance detection model with neural models to deal with this mismatch. Experiment results confirmed the limitations of our original pipeline and the improved generalizability of the modified pipeline, which will be covered in Section 4. We also experimented with other alternative techniques for automatic runs such as extracting passages from documents using a sentence classifier that took as input the entire document using a BigBird transformer [9]. This technique ensured the entire document context was taken into account when selecting sentences for query-dependant extracted passages. For answer prediction, we experimented with a technique that averaged the transformer outputs of the top-ranked 16 documents as a way to determine the correct answer for a given query.

We also produced manual runs for both tasks. For answer prediction, the first two authors manually used Google to find relevant documents and determined the correct answers based on those top-ranked results along with their perceived credibility. Note that the production of this run was independent of the track's organizers to avoid cheating. For web retrieval, the first two authors manually assessed passages selected by neural models from top documents ranked by neural rerankers. For each topic, the aim was to find at least ten correct (based on the answer prediction run) and useful documents from credible sources and to sort them in the order of preference.

The results show that our previous logistic regression model still worked well as an automatic answer prediction technique, achieving an AUC score of 0.864 and an accuracy of 70%. Compared with the new top-16-document aggregator, the accuracy and AUC score of our previous method were much better, which indicates that looking at more documents using a simple approach may be more effective than looking at fewer documents using a complex approach. For the web retrieval task, our previous pipeline could improve the compatibility difference score to 0.076 from Mono-T5 reranked results with a score of 0.052. Our new BigBird passage extraction paired with the Mono-T5 reranker achieved a compatibility difference score of 0.089, beating our previous pipeline. Finally, as expected, our manual runs achieved the best performance among our runs in both tasks, much better than our automatic runs.

2 Methods and Materials

In general, our methods were composed of several stages, including initial retrieval, passage extraction, neural reranking, stance detection, answer prediction, and final reranking based on predicted answers.

2.1 Health Misinformation Track

Since 2019, the TREC Health Misinformation Track has been focusing on general consumer health questions, with a small divergence in 2020 which focused on COVID-related health questions. In 2022, this track had two tasks: web retrieval and answer prediction.

- **Answer Prediction**: Participants were asked to figure out answers (yes or no) to the provided 50 health questions. Runs were evaluated using classic classification metrics: True Positive Rate (TPR), False Positive Rate (FPR), Accuracy, and Area Under the Curve (AUC).
- **Web Retrieval**: Participants were asked to retrieve useful and correct documents based on their predicted answers from the **Answer Prediction** task. Runs were evaluated using the Compatibility metric [5].

Topics and qrels from the 2019 track and 2021 track were used as training examples for our models.

2.2 Stage 1: Initial Retrieval

We used Anserini's BM25 to retrieve the initial set of relevant documents. For each topic, we retrieved the top 1000 documents using the default parameters of $k_1 = 0.9$ and $b = 0.4$. The index was built from the entire `c4.noclean` collection using Anserini's IndexCollection program with its default English analyzer which used Apache Lucene's (v8.0) implementations of the standard tokenizer, Porter stemmer, and some typical text cleansing techniques such as stop word filtering and lowercase conversions.

2.3 Stage 2: Passage Extraction and Neural Reranking

One of the key components in this pipeline was the method for passage extraction. Modern solutions to question answering tasks include the use of BERT-based transformers. However, web documents are normally much longer than the typical 512-token limit of most transformer models.

In previous years, a fairly common approach was to extract short passages and feed them into transformer models [7]. For example, a web document was first divided into sliding windows of six sentences with steps of three sentences. Each passage (window of six sentences) was then independently fed into the Mono-T5 [6] transformer model. After obtaining the relevance score for each of these passages from Mono-T5, the top-scoring passage was selected as the extracted passage to represent the document. Mono-T5 is a T5-based transformer model that was fine-tuned on the MS-MARCO passage retrieval dataset[2] for relevance reranking. Extracting passages in this manner has generally been shown to be effective in text retrieval. In practice, this method has two shortcomings. Firstly, it is computationally expensive. For a document with 20 passages, we need to run a transformer model 20 times to get an extracted passage. Secondly, relevant information to a query may be spread across a web page in nonconsecutive text spans. Only one continuous span for the extracted passage probably won't contain all relevant information from a document.

Another method was to use domain-specific heuristics to extract relevant sentences from a document. In the 2021 track, we used predefined keywords to determine whether a sentence should appear in the extraction [10]. Since queries are about health issues and treatments, certain words could be good indicators of relevance. We scored each sentence based on the frequencies of words such as "dangerous" and "effective" as well as query terms. While this approach worked well previously, its generalizability was limited.

As mentioned, web documents tend to be longer than the 512-token limit commonly seen in transformer models. And answers to questions can also come from multiple non-consecutive sentences. To obtain a more powerful passage extraction model this year, we fine-tuned the BigBird transformer model [9] on the MASH-QA dataset [11]. The MASH-QA dataset was specifically designed for this problem. It is a question answering dataset in the medical domain where answers need to be extracted from multiple non-consecutive parts within each document. Documents in this dataset come from WebMD[1], and questions are from users of this site. The answers are excerpts selected by medical experts from documents on this website. Labeled relevant text spans are nonconsecutive and spread over each document. The BigBird [9] model has a higher token-length limit of 4096 instead of the typical 512-token limit. We split each document into sentences using spacy[2] and prefixed each sentence with a special token. We then added a linear classifier on top of each special token at the final layer and used the final layer representation of that token to classify the sentence as relevant to answering the question or not. In this way, when classifying sentences, we take into account the context of the entire document rather than classifying each sentence in isolation. Classifying sentences individually without the surrounding context was found to have subpar performance for question answering and passage extraction tasks [11]. During inference, if there still were documents longer than 4096 tokens, we simply split those documents into multiple chunks.

As the last step at this stage, we reranked those extracted passages using the Mono-T5 model. The obtained ranked list of passages was utilized in the next question answering module in our pipeline.

2.4 Stage 3: Stance Detection and Answer Prediction

With the retrieved information from the last stage, we could build a model to get answers from those relevant passages with respect to those health-related questions. Specifically, given a relevant passage and a question (with a yes or no answer), this module needs to predict the answer indicated by the passage. We concatenated the question and the relevant passage together and fed them into the transformer. Then the final layer representation of the prepended `[CLS]` token was fed into a nonlinear classifier. We finetuned this model on the 2019 qrels by converting the document-level supportiveness judgments into yes or no answers.

We applied this module to predict the answer for every retrieved document. Then we could figure out the correct answer to the health question by looking at answers from those top-ranked documents.

For each topic, we took the top 16 passages from Mono-T5 and prepended each passage with the question. Besides, we prepended with some auxiliary features. One auxiliary feature was a special `[HON]` token if the host has a HONcode certificate. HONcode was a non-profit organization that handed out certificates to websites if they followed an 8-point code of conduct that promoted principles such as transparency, attribution, confidentiality, etc. We also prepended documents with their hostnames. These two features were found to yield a slight bump to the compatibility metric on previous years' data.

Finally, we fed the 16 query-passage pairs with those two additional features to a BERT-based classifier and averaged the output (logits). If the final result was greater than zero, the answer was predicted to be "yes". Otherwise, the answer was predicted to be "no". To train this answer prediction model, we used the White and Hassan topics provided by Zhang et al. [10], which was originally from White and Hassan [8]. We used the 2019 topics and the 2021 topics as development sets when tuning the model.

2.5 Stage 4: Final Reranking

At the final reranking stage, we needed to rerank documents based on their levels of agreement with our predicted answers, i.e., their correctness. From the last stage, we had a yes/no logit for each query-document pair. We then combined it with the document's Mono-T5 relevance score using the following formula:

$$S'_{q,d} = \frac{2S_{q,d}}{1 + e^{\alpha \cdot L_{q,d} \cdot \bar{L_q}}}$$

where $S_{q,d}$ is the original Mono-T5 score, $S'_{q,d}$ is the new ranking score, $L_{q,d}$ is the the yes/no logit for the document d, α is a hyperparameter, and $\bar{L_q}$ is the answer prediction logit. The intuition behind this formula is that documents that are heavily in disagreement with our predicted answer will have their score suppressed near 0, thus disappearing from the top of the final ranked list. Meanwhile, documents that are in agreement will have their scores boosted up to 100%.

2.6 Manual

In addition to the automatic methods mentioned above, we also performed manual assessments to figure out correct answers and find high-quality documents in terms of their usefulness, correctness, and credibility.

For the answer prediction task, the first two authors, which we'll refer to as *assessors*, independently determined their perceived answers to all those 50 topics, using search engines like Google to find credible evidence sources, such as `healthline.com` and `webmd.com`. Then, they discussed and reached an agreement on the final answers to those 50 topics.

[1] `https://www.webmd.com/`

[2] `https://spacy.io/`

For the web retrieval task, the assessors manually judged the passages (six sentences) selected by Mono-T5 from top documents ranked by Mono-T5, with respect to usefulness, correctness, and credibility. Correct documents here mean that their stances align with our perceived answers from above. The idea is that Mono-T5 is good at finding relevant passages from the document and assessing passages is much faster than assessing the full documents, though at the cost of potentially lower accuracy. Each of the assessors worked on 25 topics to find at least 10 most useful and correct documents for each topic.

Two passes were performed over the documents and each pass followed the order of documents ranked by Mono-T5. In the first pass, assessors primarily focused on credibility. If the source seemed credible (well-known credible or HONCode-certified[3]), the assessors then further judged its usefulness and correctness. If the content was also useful and correct, then a **very useful and correct** document was found. Assessors kept finding and judging until they reached the 200th document or found 10 very useful and correct documents. If they did not find 10 very useful and correct documents after looking through the top 200 documents, they would start over and focus on the usefulness and correctness instead until they found 10 **useful and correct** documents (including those found in the first pass). After those two passes, all documents were then ranked in the following order: very useful and correct, useful and correct, and not judged. Documents in the same class were ranked by their scores from Mono-T5. Finally, for each topic, assessors performed a preference ordering of the top 10 documents based on their personal judgments of which the best document was, which the second best document was, etc.

3 Submitted Runs

Some of our submitted runs reused our previous approach [10] with minor modifications, while others were new and have been described in Section 2.

For the auxiliary task: answer prediction, we produced the following runs:

- `WatS-AP-Baseline` [automatic]:
  Basically, the same pipeline from our previous work [10] using the `question` field, with minor modifications in the stance detection model to adapt to the topic format shift.

- `WatS-AP-Baseline-L1` [automatic]:
  The difference to `WatS-AP-Baseline` was that L1 regularization was enabled in the logistic regression model.

- `WatS-AP-MT5` [automatic]:
  The difference to `WatS-AP-Baseline` was that the intuitive sentence selection algorithm was replaced by Mono-T5 and one additional reranking stage was added after the first retrieval using Mono-T5.

- `WatS-AP-MT5-L1` [automatic]:
  The difference to `WatS-AP-MT5` was that L1 regularization was enabled in the logistic regression model.

- `WatS-BB75-MT5-TA` [automatic]:
  We used passages extracted by BigBird and aggregated the top 16 passages along with their auxiliary features to predict a yes/no answer to the question.

- `WatS-AP-Manual` [manual]:
  Manual determination of the correct answers using relevant documents returned by Google.

For the core task: web retrieval, we produced the following runs:

- `WatS-Query` [automatic]:
  The `query` field was used as the query for BM25 retrieval.

- `WatS-Question` [automatic]:
  The `question` field was used as the query for BM25 retrieval.

- `WatS-MT5-MT5` [automatic]:
  We reranked `WatS-Question` by extracting passages from documents using spans of 6 sentences with steps of 3 sentences. These passages along with the question are scored with Mono-T5 and the documents are reranked using their best-scoring passages.

- `WatS-Trust` [automatic]:
  The same pipeline from our previous work using the `question` field, with minor modifications in the stance detection model to adapt to the topic format shift. This run was based on predicted answers from `WatS-AP-Baseline`.

- `WatS-Trust-L1` [automatic]:
  The difference to `WatS-Trust` was that this run was based on predicted answers from `WatS-AP-Baseline-L1`.

- `WatS-Trust-MT5` [automatic]:
  The difference to `WatS-Trust` was that the

[3] `https://myhon.ch/en/`

heuristic sentence selection algorithm was replaced by Mono-T5 and one additional reranking stage was added after the first retrieval using Mono-T5. This run was based on predicted answers from `WatS-AP-MT5`.

- `WatS-Trust-MT5-L1` [automatic]:
  The difference to `WatS-Trust-MT5` was that this run was based on predicted answers from `WatS-AP-MT5-L1`.

- `WatS-Bigbird2_75-MT5` [automatic]:
  For passage extraction, we used BigBird and classified each sentence as relevant to the question or not. Sentences with classification scores higher than 75% were included in the document passage. Documents with no relevant sentences were discarded. Documents were reranked based on the Mono-T5 scores of these extracted passage and question pairs.

- `WatS-Bigbird2_75-MT5-TA1` [automatic]:
  We reranked `WatS-Bigbird2_75-MT5` using our answer prediction model with an α of 0.2.

- `WatS-Bigbird2_75-MT5-TA2` [automatic]:
  Same as `WatS-Bigbird2_75-MT5-TA1` except that α is 0.1.

- `WatS-Manual` [manual]:
  As is described in Section 2.6.

4 Results

In this section, we first analyze our runs for the answer prediction task, which our web retrieval runs depend upon, and then analyzed runs for the web retrieval task.

4.1 Answer Prediction

Table 1 shows the classification performance of our answer prediction runs. From the comparison between `AP-Baseline` and `AP-MT5` and the comparison between `AP-Baseline-L1` and `AP-MT5-L1`, we can see that replacing the heuristic sentence selection algorithm with Mono-T5 and adding an additional reranking stage after the initial retrieval indeed led to obvious improvements in predicting correct answers. This confirms our intuition that the sentence selection method lacks generalizability which may result in suboptimal performance of the stance detection model and therefore the final answer prediction performance.

Run	**TPR**	**FPR**	**Acc**	**AUC**
`AP-Baseline`	0.800	0.680	0.560	0.557
`AP-Baseline-L1`	0.720	0.640	0.540	0.565
`AP-MT5`	0.960	0.680	0.640	0.813
`AP-MT5-L1`	1.000	0.600	0.700	0.864
`BB75-MT5-TA`	0.440	0.120	0.660	0.691
`AP-Manual`	0.880	0.000	0.940	0.940

Table 1: Answer Prediction Task: Classification Performance. TPR: True Positive Rate, FPR: False Positive Rate, Acc: Accuracy, AUC: Area Under the Curve. Due to the space limit, the common prefix `WatS-` of those runs is omitted in this table.

Run	**C(help)**	**C(harm)**	**C(△)**
`BM25-Query`	0.171	0.140	0.031
`BM25-Question`	0.193	0.149	0.044
`Bigbird2_75-MT5`	0.217	0.209	0.007
`Bigbird2_75-MT5-TA1`	0.244	0.171	0.073
`Bigbird2_75-MT5-TA2`	0.242	0.153	0.089
`MT5-MT5`	0.246	0.194	0.052
`Trust`	0.205	0.142	0.063
`Trust-L1`	0.187	0.153	0.034
`Trust-MT5`	0.245	0.188	0.056
`Trust-MT5-L1`	0.253	0.177	0.076
`Manual`	0.284	0.140	0.145

Table 2: Web Retrieval Task: Overall Performance Using the Compatibility Metric [5]. C(help): helpful Compatibility score, C(harm): harmful Compatibility score, C(△): difference between the helpful Compatibility score and the harmful Compatibility score. Due to the space limit, the common prefix `WatS-` of those runs is omitted in this table.

`BB75-MT5-TA` had relatively poor accuracy and AUC, demonstrating that the top 16 documents may not be sufficient for deriving answers.

4.2 Web Retrieval

From Table 2, we can observe that none of our automatic methods could outperform the manual run, in terms of the helpful compatibility score and the compatibility difference (main metric).

Comparing `Trust` and `Trust-MT5-L1`, it is interesting to see that those two runs were not so different in terms of the compatibility difference, even though the answer prediction run `AP-MT5-L1` that `Trust-MT5-L1` was based on performed way better than `AP-Baseline` that `Trust` was based on.

Our new BigBird passage extraction paired with a Mono-T5 reranker had a relatively poor helpful compatibility score compared to using Mono-T5 for both passage extraction and reranking (0.217 v.s. 0.246). However, our reranking using BigBird and Mono-T5 produced our best automatic run, in terms of the main metric - compatibility difference.

5 Discussion

During our manual assessment of Mono-T5 selected passages from Mono-T5 reranked documents, we noticed a lot of nearly identical passages, some of which were from spam web pages that directly copied contents from other sources. We expect a better quality of retrieved documents had we also performed spam filtering on the document collection or on the returned documents from the first-stage retrieval.

Retrospectively, we did not properly use tools like Mono-T5 when producing the manual run. We should modify the query to include the answers so that the retrieval algorithm can find more documents aligned with our answers rather than documents that oppose them, which may help us find more correct documents or at least save our efforts to assess more documents further down the list of retrieved documents.

As mentioned, our BigBird aggregator failed to predict answers as well as our previous logistic regression aggregator. Especially in topics where the top results were mostly incorrect, this approach would do poorly. However, even with relatively poor quality of answer prediction (AUC: 0.691), our reranker still managed to promote correct information and reduce misinformation, improving the compatibility difference from 0.007 to 0.089. To some extent, the reranking compensated the question answering module. If we were to combine the BigBird passage extraction with the more effective logistic regression-based answer prediction, we might have an effective question answering and reranking pipeline with relatively less expensive computational costs.

6 Future Work

With more complex models being released in the field of NLP, we expect to see more improvements if the language models used in our methods are replaced with these larger and more powerful models. Besides, improving the ratio of correct documents retrieved with BigBird and combining it with our logistic regression-based answer prediction model can be an effective approach, while being relatively computationally efficient compared with other methods proposed in previous years' tracks.

7 Conclusion

In this paper, we described our participation in the TREC 2022 Health Misinformation Track. For the answer prediction task, our previous method worked fairly well compared with new methods proposed in this paper. We believe that simple aggregation techniques that look at more relevant documents are likely to work better than more complex aggregators looking at fewer relevant documents. For the web retrieval task, our manual run outperformed our automatic runs by a great margin, though it could be even better if we performed manual assessments in a more structured and consistent way. Our new automatic method achieved better performance than our previous approaches, showing the potentials of document-aware sentence-level passage extraction, which still needs further refinement to reach its maximum effectiveness for this task.

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (RGPIN-2020-04665, RGPAS-2020-00080), and in part by the Digital Research Alliance of Canada.

References

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

[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268* (2016).

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

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

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

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

[7] Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021. 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*. 2066–2070.

[8] Ryen W White and Ahmed Hassan. 2014. Content Bias in Online Health Search. *ACM Transactions on the Web (TWEB)* 8, 4 (2014), 1–33.

[9] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big Bird: Transformers for Longer Sequences. *Advances in Neural Information Processing Systems* 33 (2020).

[10] Dake Zhang, Amir Vakili Tahami, Mustafa Abualsaud, and Mark D. Smucker. 2022. 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*. 2099–2104.

[11] Ming Zhu, Aman Ahuja, Da-Cheng Juan, Wei Wei, and Chandan K. Reddy. 2020. Question Answering with Long Multiple-span Answers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 3840–3849.