Webis at TREC 2022: Deep Learning and Health Misinformation

Alexander Bondarenko *
Friedrich-Schiller-Universität Jena

Maik Fröbe *
Friedrich-Schiller-Universität Jena

Lukas Gienapp *
Leipzig University

Alexander Pugachev *
HSE University

Jan Heinrich Reimer *
Martin-Luther-Universität Halle-Wittenberg

Ferdinand Schlatt
Martin-Luther-Universität Halle-Wittenberg

Ekaterina Artemova
Ludwig-Maximilians-Universität München

Martin Potthast
Leipzig University and ScaDS.AI

Benno Stein
Bauhaus-Universität Weimar

Pavel Braslavski
HSE University
Ural Federal University

Matthias Hagen
Friedrich-Schiller-Universität Jena

ABSTRACT

We describe the Webis group's participation in the TREC 2022 Deep Learning and Health Misinformation tracks. Our runs submitted to the Deep Learning track focus on improving the pairwise retrieval model duoT5 by combining a greedy aggregation algorithm with document augmentation. In the Health Misinformation track, our submissions to the Answer Prediction task exploit pre-trained question answering and claim verification models, whose input is extended by evidence information from PubMed abstracts. For the Web Retrieval task, we explore re-ranking based on the closeness of the predicted answers for each web document in the initial ranking to the predicted "true" answer of a topic's question.

1 INTRODUCTION

We participated in two TREC 2022 tracks: Deep Learning and Health Misinformation. As for the Deep Learning track, with our four runs we investigate whether the default aggregation of pairwise preferences in duoT5 can be further improved. The default implementation of duoT5 already is very effective by deriving and sym-sum-wise aggregating pairwise preferences for all possible pairs of documents. We investigate whether a different greedy aggregation scheme or whether deriving each pairwise preference multiple times using perturbed variants of the query or the documents can help. Our results show that a greedy aggregation improves the effectiveness of duoT5 substantially, but calculating the pairwise preferences multiple times with perturbations does not improve the effectiveness.

As for the Health Misinformation track, in our 20 runs (10 for each task) we use several pre-trained question answering (QA) and scientific claim verification systems to predict a "correct" yes/no answer to a topic's question. As input to the systems, we use PubMed abstracts to add evidence information (context) from trustworthy sources to the questions. The predicted answer is then used as an estimated true answer to construct rankings where documents are simultaneously sorted by their topical relevance and the predicted correctness of the contained information.

2 DEEP LEARNING TRACK

We submitted the results of four approaches to the TREC Deep Learning track. All four systems are implemented in PyTerrier [16] where we first re-rank the task's official top-100 document candidates using monoT5 [19]. Then, we calculate duoT5 preference scores for all pairs (sometimes in multiple variants) of documents in the top-50 of the monoT5 ranking and compare the official duoT5 aggregation of the duoT5 ranking with a greedy aggregation algorithm. Two variants calculate for each document pair multiple pairwise preferences where we create augmentations of query-document-document triples (e.g., replacing the original query with queries generated via docT5query [17] pre-calculated by Ma et al. [14]). We use ir_datasets [15] to access the passages for re-ranking. We used existing models from Hugging Face for MonoT5[1] and DuoT5[2] which are trained on version 1 of MS MARCO (i.e., we do not train models). Using models trained on version 1 is recommended [5] (e.g., version 2 has more noisy labels [6]). Calculating all pairwise preferences (including all augmentations) took roughly 5 hours on a single core of an A100 GPU.

We submitted four runs out of which three were pooled and one is the baseline:

Webis-dl-duot5. We re-rank the top-100 candidates of the official baseline with monoT5 and re-rank the top-50 of the monoT5 ranking with duoT5. We use the implementation of monoT5 and duoT5 in PyTerrier with the models trained on version 1 of MS MARCO mentioned above.

Webis-dl-duot5-g. We re-rank the top-100 candidates of the official baseline with monoT5 and re-rank the top-50 of the monoT5 ranking using duoT5 and greedy aggregation. We use the implementation of monoT5 and duoT5 in PyTerrier with the models

*These authors contributed to the paper equally and are listed alphabetically.

TREC 2022, November 14–18, 2022, Gaithersburg, Maryland

[1] https://huggingface.co/castorini/monot5-3b-msmarco
[2] https://huggingface.co/castorini/duot5-3b-msmarco

Table 1: **The effectiveness of our four runs for the re-ranking scenario in the TREC Deep Learning track. We report nDCG@10 and the mean reciprocal rank (MRR).**

Run	nDCG@10	MRR
Webis-dl-duot5	0.497	0.711
Webis-dl-duot5-g	0.531	0.823
Webis-dl-duot5-aug-1	0.493	0.713
Webis-dl-duot5-aug-2	0.489	0.648

trained on version 1 of MS MARCO mentioned above. To aggregate the pairwise preferences into retrieval scores, we use a greedy approach proposed by Cohen et al. [4], as previous experiments showed that greedy aggregation is more effective than the default sym-sum aggregation [9]. The greedy aggregation algorithm is proven to closely approximate the best total order in terms of the number of violated preferences [4].

Webis-dl-duot5-aug-1. We re-rank the top-25 results of our webis-dl-duot5-g run by aggregating multiple pairwise preferences obtained via duoT5 on augmented document pairs. Out of 9 augmentation patterns ((1) no augmentation, (2, 3, 4) using only the first one, two, or three sentence(s) of the passages, and (5, 6, 7, 8, 9) expand the passages with a query obtained via docT5query), two methods to aggregate pairwise scores (greedy and sym-sum), and five methods to aggregate augmented scores (min, max, mean, median, sum), we selected the combination with the highest nDCG@10 on the TREC 2020 DL data. This hyperparameter optimization yielded an approach that used five augmentations ((1) no augmentation, (2) both passages in a comparison shortened to the first two sentences, (3-5) variants of passage expansions) that are aggregated into a single pairwise score using min aggregation, and the pairwise scores are aggregated into retrieval scores using greedy aggregation.

Webis-dl-duot5-aug-2. We re-rank the top-25 results of our webis-dl-duot5-g run by aggregating multiple pairwise preferences obtained via duoT5 on augmented document pairs. Out of 9 augmentation patterns ((1) no augmentation, (2, 3, 4) using only the first one, two, or three sentence(s) of the passages, and (5, 6, 7, 8, 9) expand the passages with a query obtained via docT5query), two methods to aggregate pairwise scores (greedy and sym-sum), and five methods to aggregate augmented scores (min, max, mean, median, sum), we selected the combination with the highest MRR on the TREC 2020 DL data. This hyperparameter optimization yielded an approach that used six augmentations ((1) no augmentation, (2,3) both passages in a comparison shortened to the first one and first three sentences, (4-6) variants of passage expansions) that are aggregated into a single pairwise score using sum aggregation, and the pairwise scores are aggregated into retrieval scores using greedy aggregation.

2.1 Evaluation

Table 1 shows the effectiveness of our four approaches in terms of nDCG@10 and the mean reciprocal rank (MRR). We follow the recommendation of the organizers of the shared task and report evaluation results with duplicate documents (even when not removing duplicates might come with disadvantages [7, 8]). Both augmentation runs decrease the effectiveness of duoT5. However, the greedy variant substantially improves upon the original duoT5.

3 HEALTH MISINFORMATION TRACK

In our 10 runs submitted to the Answer Prediction task, we use various pre-trained QA and claim verification systems to infer correct answers to the yes/no health questions from 50 topics by aggregating the answers predicted for the top-k PubMed abstracts retrieved when using a topic's `question` field as a query. We also use the predicted answers as candidate "true" answers in our 10 runs submitted to the Web Retrieval task. In our ranking approaches, we order documents by combining document topical relevance with the predicted correctness of the contained information.

3.1 Answer Prediction Task: Our Runs

To predict a "correct" answer to the 50 yes/no health questions like "Are vaccines linked to autism?", we test pre-trained question answering and claim verification models. As input to the models, we use a topic's `question` field and relevant evidence information extracted from trustworthy sources like PubMed.[3]

Runs using QA models. For each topic, we first retrieve 20 or 1000 PubMed abstracts as evidence candidates by submitting topics' questions to one of the following retrieval systems: (1) PubMed API,[4] (2) Google Custom Search API,[5] or (3) BM25 retrieval [21] (Elasticsearch[6] implementation) on an index of 33.5 million PubMed abstracts.[7] For each question–abstract pair, we then let a QA model predict an answer score between 0 (no) and 1 (yes). As QA models, we use (1) a BioLinkBERT-large model [25] pre-trained on PubMedQA [11],[8] (2) a RoBERTa-BoolQ-base model [13] pre-trained on the BoolQ dataset [2],[9] and (3) a UnifiedQA-T5-large model [12] pre-trained on various question answering datasets.[10] We do not fine tune the models. As the final answer score for the topics' questions (in the range from 0 to 1), our runs use different ways of aggregating the predicted answer scores for each evidence document. Following the task requirements, we include in each run a numerical score and a binary yes/no answer label (using a decision threshold of 0.5). Our submitted runs are:

Webis-goo-boolq-abs. For each task topic, we use the `question` field as a query to the Google Custom Search API (limited to a search in PubMed). For each of the up top-20 returned PubMed abstracts, the pre-trained RoBERTa-BoolQ-base model predicts the probability of whether the abstract is on the 'yes'-side of an answer. The final answer prediction score for a topic is the average of the individual answer scores across all abstracts.

Webis-goo-lbert-abs. Analogous to the previous run, but using the pre-trained BioLinkBERT-large QA model.

[3] https://pubmed.ncbi.nlm.nih.gov/
[4] https://www.ncbi.nlm.nih.gov/books/NBK25499/#chapter4.ESearch
[5] https://developers.google.com/custom-search/v1/using_rest
[6] https://www.elastic.co/
[7] Obtained from https://pubmed.ncbi.nlm.nih.gov/download/
[8] https://huggingface.co/Tejas21/bio-linkbert-large_pubmedqa-hf
[9] https://huggingface.co/luztraplet/roberta-large-finetuned-boolq
[10] https://huggingface.co/allenai/unifiedqa-t5-large

Webis-goo-lbert-title-abs. Analogous to the previous run, but prepending a returned abstract's title to the abstract content before passing to the QA model.

Webis-nlm-boolq-abs. Similar to the first run, but using the native PubMed search API as the retrieval system and a topic's `query` fields as queries (since the API does not process well natural language questions). For topics where no or only one result is retrieved, we reformulated the query by hand (e.g., using synonyms) until at least two abstracts are found.

Webis-nlm-lbert-abs. Analogous to the previous run, but the pre-trained BioLinkBERT-large QA model is used.

Webis-uniqa-dis. For this run, we use BM25 to retrieve 1000 abstracts from the PubMed abstracts index (topic `question` fields are used as queries). Then, we re-rank the results with monoT5 [19] and again re-rank the top-50 results (of the first re-ranking step) with duoT5 [19].[11] We use PyTerrier [16] to implement re-ranking. The predicted answer scores returned for 1000 question–abstract pairs using the UnifiedQA-T5-large model are aggregated in the topic's final answer prediction score by discounting ranking positions, assuming that answers from higher ranked (i.e., more relevant) abstracts might be closer to the true answer and thus should contribute more to the final topic's answer score.

We aggregate the topic answer score as follows: (1) Given the predicted answer scores score_i for the abstract at rank i we compute the discounted cumulative answer score DCA for top-k documents:

$$\text{DCA}_k = \sum_{i=1}^{k} \frac{\text{score}_i}{\log_2 i + 1}$$

Then, (2) the normalization factor is computed as the maximum achievable (ideal) discounted cumulative answer score IDCA:

$$\text{IDCA}_k = \sum_{i=1}^{k} \frac{1}{\log_2 i + 1}$$

Finally, (3) we use the normalized discounted cumulative answer score nDCA_k (with $k = 1000$) as the prediction score for the questions from each topic:

$$\text{nDCA}_k = \frac{\text{DCA}_k}{\text{IDCA}_k}$$

Runs using claim verification models. For each topic's question, we first retrieve 1000 abstracts from the index of 33.5 million PubMed abstracts using BM25 (topic `question` fields are used as queries). Then, we re-rank the top-1000 results with monoT5 followed by re-ranking with duoT5 the top-50 results from the first re-ranking step (analogous to the *Webis-uniqa-dis* run). For each question–abstract pair, we collect predicted answer scores returned by the claim verification model LongChecker [23, 24] pre-trained on the FEVER dataset [22][12] or the Vera model [18] pre-trained on the data from the TREC 2019 Health Misinformation track [1]. We do not fine-tune the models. Even though the models were originally trained to predict the support/refute probabilities (given a claim and a text passage), we take these predictions as yes/no answer prediction scores. The final answer prediction score for each topic is calculated using nDCA (as described above) aggregated for 1000 question–abstract pairs.

Webis-longck-dis. After the retrieval and re-ranking steps, the answer is predicted by aggregating the prediction scores returned by LongChecker (using questions and abstracts plus titles as a context input) with a ranking position discount (nDCA_{1000}, analogous to the run *Webis-uniqa-dis*).

Webis-verasent-dis. Analogous to the previous run, but using the Vera model for answer prediction. Since Vera has a 512-token input limitation, we use as input prompt only the "most relevant" sentences from abstracts found using heuristics proposed by Zhang et al. [26] that use a handcrafted list of indicator words.

Runs using a combination of QA and claim verification. For the following runs, we use the same retrieval (BM25) and re-ranking steps (monoT5 and duoT5) and average two prediction scores returned by UnifiedQA and LongChecker. The final answer prediction for each topic is again aggregated using nDCA_{1000} from the individual answer scores for the 1000 question–abstract pairs.

Webis-longck-uniqa-dis. For this run, after re-ranking, the final score is calculated from the averaged UnifiedQA (only abstract as context input) and LongChecker (abstract and title as context input) predictions discounted by the ranked positions.

Webis-longck-uniqa-ax-dis. Analogous to the previous run, but additionally top-1000 PubMed abstracts are also axiomatically re-ranked [10] based on their publication date to resolve potentially contradicting information from different publications. This way, the final predicted answer scores are more influenced by the more recently published abstracts.

Table 2: The Health Misinformation track's answer prediction results provided by the organizers (sorted by the AUC scores or by the next metric in case of ties). Reported are AUC and accuracy scores and false and true positive rates.

Run	AUC	Acc.	FPR	TPR
Webis-verasent-dis	0.81	0.70	0.40	0.80
Webis-longck-dis	0.79	0.64	0.36	0.64
Webis-nlm-boolq-abs	0.69	0.52	0.96	1.00
Webis-longck-uniqa-dis	0.66	0.62	0.48	0.72
Webis-uniqa-dis	0.66	0.62	0.48	0.72
Webis-longck-uniqa-ax-dis	0.66	0.60	0.48	0.68
Webis-goo-boolq-abs	0.65	0.52	0.96	1.00
Webis-goo-lbert-abs	0.48	0.50	0.88	0.88
Webis-goo-lbert-title-abs	0.48	0.50	0.92	0.92
Webis-nlm-lbert-abs	0.48	0.50	0.80	0.80
Median all participants	*0.71*	*0.64*	*0.48*	*0.80*

3.2 Answer Prediction Task: Evaluation

The results for our runs submitted to the Answer Prediction task as provided by the track organizers are reported in Table 2. Overall, we observe that discounting the answer prediction scores based on the rank of retrieved evidence documents more often than simple

[11] https://huggingface.co/castorini/monot5-3b-med-msmarco, https://huggingface.co/castorini/duot5-3b-med-msmarco
[12] Using fever_sci checkpoints https://github.com/dwadden/multivers

averaging yields higher AUC and accuracy scores. Similarly, using claim verification systems is more accurate than QA systems for predicting an answer. These differences however might also be caused by the retrieval approaches used to find evidence documents (at this point we have not evaluated their retrieval effectiveness) or by the differences in datasets both types of systems were trained on. The answer predictor that is based on the Vera claim verification model that uses only the "most relevant" sentence selection from PubMed abstracts is the most accurate in predicting correct answers. However, the approach that uses LongChecker has the lowest false positive rate, worth considering because this error type is most harmful for health-related questions.

3.3 Web Retrieval Task: Our Runs

After predicting the answer to a topic's question, we retrieve documents with BM25 from one billion documents of the C4 corpus [20] that we indexed with Elasticsearch. Then, we re-rank the results with monoT5 and again re-rank the top-50 results (of the first re-ranking step) with duoT5 (the same models as in Section 3.1). The answer score for each retrieved document from C4 is then predicted in a similar way as described in Section 3.1. To combine a document's answer score with the retrieval score for the final ranking, we first calculate the difference of the predicted answer scores between the topic T (predicted "true" answer) and each document D:

$$\Delta\text{answer} = |\text{answer}(T) - \text{answer}(D)|$$

Our runs further use the closeness $1 - \Delta\text{answer}$ to the predicted "true" topic answer to boost the initial retrieval scores.

Runs with a linear score boosting. For each run in this group, we use BM25 to retrieve 1000 abstracts from the PubMed abstracts index (topic `question` fields are used as queries). Then, we re-rank the results with monoT5 and again re-rank the top-50 results (of the first re-ranking step) with duoT5 using PyTerrier [16]. Answer conflicts are resolved with axiomatic re-ranking (more recently published abstracts are ranked higher). For each of the 1000 question–abstract pairs, we collect answer prediction scores returned by pre-trained claim verification and/or QA models for each retrieved abstract and then aggregate the final topics' answer score by discounting ranking positions with nDCA_{1000}. We then retrieve 1000 documents from C4 using Elasticsearch's BM25, re-rank with monoT5, and the top-50 of the first re-ranker with duoT5. Document answer scores are predicted with the same claim verification and QA models as in Section 3.1. Using the aggregated topic answer score and the individual answer scores for each retrieved C4 document, we boost the retrieval score linearly based on the closeness between a re-ranked document's answer score and the predicted topic answer:

$$\text{score}_{\text{lin}}(D) = \text{score}_{\text{duoT5}}(D) * (1 - \Delta\text{answer})$$

Webis-longck-ax-lin. Predict answer scores using the pre-trained claim verification model LongChecker[13] with abstract texts, abstract titles, and document text as a context input.

Webis-uniqa-ax-lin. Predict answer scores using the pre-trained QA model UnifiedQA-T5-large[14] with abstract texts and document text as context input.

[13] Using fever_sci checkpoints from https://github.com/dwadden/multivers
[14] https://huggingface.co/allenai/unifiedqa-t5-large

Webis-longck-uniqa-ax-lin. Predict answer scores using the averaged UnifiedQA and LongChecker scores (abstract texts, abstract titles (only LongChecker), and document text as a context input).

Runs with a polynomial score boosting. Using the aggregated topic answer score and the individual answer scores for each retrieved from C4 document, we boost retrieval scores as follows:

$$\text{score}_{\text{pol}}(D) = \text{score}_{\text{duoT5}}(D) * (1 - \Delta\text{answer}^2)$$

Webis-longck-ax-pol. Predict answer scores for abstracts and documents using the pre-trained claim verification model LongChecker (fever_sci checkpoints, again) with abstract texts, abstract titles, and document text as a context input.

Webis-uniqa-ax-pol. Predict answer scores using the pre-trained QA model UnifiedQA-T5-large (abstract texts and document text as a context input).

Webis-longck-uniqa-ax-pol. Predict answer scores using the averaged UnifiedQA and LongChecker scores (abstract texts, abstract titles (only LongChecker), and document text as a context input).

Webis-longck-uniqa-pol. Analogous to the previous run, but without axiomatic re-ranking by the publication date.

Runs with a weighted score combination. Using the aggregated topic answer score and the individual answer scores for each retrieved document, we combine a C4 document retrieval score with the closeness of that document's answer score to the predicted topic answer weighted by a trade-off α:

$$\text{score}_{\text{com}}(D) = \alpha \cdot \text{score}_{\text{duoT5}}(D) + (1 - \alpha) \cdot (1 - \Delta\text{answer})$$

Webis-longck-ax-com. Predict answer scores with the pre-trained claim verification model LongChecker (fever_sci checkpoints, again) with abstract texts, abstract titles, and document text as a context input ($\alpha = 0.75$).

Webis-uniqa-ax-com. Predict answer scores using the pre-trained QA model UnifiedQA-T5-large (abstract texts and document text as a context input; $\alpha = 0.75$).

Webis-longck-uniqa-ax-com. Using the averaged UnifiedQA and LongChecker answer scores (abstract texts, abstract titles (only LongChecker), and document text as a context input; $\alpha = 0.75$).

3.4 Web Retrieval Task: Evaluation

For the Web Retrieval task, the retrieval effectiveness of the submitted runs is evaluated using nDCG, precision, and a compatibility measure [3] w.r.t the usefulness, correctness, and helpfulness and harmfulness of documents. The results for our runs in Table 3 show that none of our approaches significantly outperforms the median retrieval effectiveness (some runs are, however, significantly worse). The runs featuring linear or polynomial score boosting (cf. runs (a)–(g) in Table 3) have significantly worse effectiveness (both nDCG and precision) on binary 'useful' and 'correct' relevance judgments as well as for graded 'usefulness' judgments. They are also significantly less compatible with 'helpful' results. Runs featuring a score combination (with trade-off $\alpha = 0.75$, cf. runs (h)–(j) in Table 3) achieve a significantly improved effectiveness (nDCG) and are significantly more compatible with 'helpful' results compared

Table 3: The Health Misinformation track's official effectiveness results for our runs. U: useful, Co: correct, Incor.: incorrect. Significant differences to other runs are marked with superscripts (Student's t-test, $p < 0.0009 = 0.05/55$, Bonferroni-corrected).

Run	Compatibility		nDCG (binary)	P@10 (binary)		nDCG (graded)
	Help	Harm	U & Co	U & Co	Incor.	Useful
(a) Webis-longck-ax-lin	0.11^{fghijk}	0.07^{eij}	$0.43^{\text{bdefghijk}}$	0.27^{hijk}	0.09^{ij}	$0.49^{\text{bcdefghijk}}$
(b) Webis-uniqa-ax-lin	0.15^{ehijk}	0.12	0.50^{aehijk}	0.31^{hijk}	0.13	0.56^{acehijk}
(c) Webis-longck-uniqa-ax-lin	0.14^{fghijk}	0.07^{eij}	0.48^{efghijk}	0.34^{hik}	0.08^{ij}	$0.52^{\text{abefghijk}}$
(d) Webis-longck-ax-pol	0.15^{hijk}	0.09	0.47^{ahijk}	0.32^{hik}	0.11	0.54^{aehijk}
(e) Webis-uniqa-ax-pol	0.18^{bhk}	0.14^{ac}	0.52^{abchijk}	0.36^{hik}	0.15	0.58^{abcdhijk}
(f) Webis-longck-uniqa-ax-pol	0.17^{achik}	0.08	0.51^{achijk}	0.38^{hk}	0.10	0.57^{achijk}
(g) Webis-longck-uniqa-pol	0.17^{achik}	0.08	0.52^{achijk}	0.38^{hk}	0.10	0.57^{achijk}
(h) Webis-longck-ax-com	0.27^{abcdefg}	0.15	0.58^{abcdefg}	0.55^{abcdefg}	0.18	0.66^{abcdefg}
(i) Webis-uniqa-ax-com	0.26^{abcdfg}	0.17^{ac}	0.58^{abcdefg}	0.52^{abcde}	0.23^{ac}	0.66^{abcdefg}
(j) Webis-longck-uniqa-ax-com	0.25^{abcd}	0.17^{ac}	0.57^{abcdefg}	0.48^{ab}	0.23^{ac}	0.65^{abcdefg}
(k) Median all participants	*0.24*	*0.13*	*0.61*	*0.53*	*0.16*	*0.69*

to the runs with linear and polynomial score boosting. Axiomatic re-ranking at the answer prediction stage does not significantly change compatibility or effectiveness (cf. runs (f) and (g) in Table 3). Compared to the runs featuring the LongChecker model [23, 24], the runs using UnifiedQA [12] or a combination of both have slightly improved effectiveness and compatibility with linear or polynomial score boosting, but the opposite effect can be observed when using a weighted score combination. Thus, whether the claim verification or QA models are better suited for the task is inconclusive. A weighted combination of the topical relevance with an answer closeness to the predicted "true" is so far the most promising.

4 CONCLUSION

In the Deep Learning track, we investigated the effectiveness of four duoT5 variants. We found that a greedy aggregation is substantially more effective than the original duoT5 at the same efficiency.

In the Health Misinformation track's Answer Prediction task, we investigated the effectiveness of pre-trained question answering and scientific claim verification models in predicting correct answers to yes/no health questions. As input to the models, we used questions and retrieved PubMed abstracts that potentially contain trustworthy evidence information. The experimental results show the investigated claim verification models to be more effective for the task than the investigated question answering models (a possible reason might be the different datasets that were originally used for training). For the Web Retrieval task, all our runs are not particularly effective (rather below the median across all evaluation metrics) but a weighted combination of the topical relevance with documents' closeness to the predicted "true" answer is significantly more effective than our linear or polynomial score boosting. As for using a predicted "true" answer during retrieval, there was no significant difference between using question answering or claim verification models even though in the Answer Prediction task the claim verification is more effective. One of the possible reasons might be that our current re-ranking approaches do not consider borderline answer predictions (close to the 0.5 answer threshold), which needs further investigation and is an interesting direction for future work.

ACKNOWLEDGMENTS

This work has been partially supported by the DFG in the project "ACQuA 2.0: Answering Comparative Questions with Arguments" (project 376430233) as part of the priority program "RATIO: Robust Argumentation Machines" (SPP 1999).

REFERENCES

[1] Mustafa Abualsaud, Christina Lioma, Maria Maistro, Mark D. Smucker, and Guido Zuccon. Overview of the TREC 2019 Decision Track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019*. NIST.

[2] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*. ACL, 2924–2936.

[3] Charles L. A. Clarke, Mark D. Smucker, and Alexandra Vtyurina. Offline Evaluation by Maximum Similarity to an Ideal Ranking. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM 2020*. ACM, 225–234.

[4] William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to Order Things. *J. Artif. Intell. Res.* 10 (1999), 243–270.

[5] N. Craswell, B. Mitra, Emine Yilmaz, and Daniel Campos. Overview of the TREC 2021 Deep Learning Track. In *Proceedings of the Thirtieth Text REtrieval Conference, TREC 2021*. NIST.

[6] Maik Fröbe, Christopher Akiki, Martin Potthast, and Matthias Hagen. Noise-Reduction for Automatically Transferred Relevance Judgments. In *Proceedings of the 13th International Conference of the CLEF Association, CLEF 2022*. Springer, 48–61.

[7] Maik Fröbe, Janek Bevendorff, Jan Heinrich Reimer, Martin Potthast, and Matthias Hagen. Sampling Bias Due to Near-Duplicates in Learning to Rank. In *Proceedings of the 43rd International ACM Conference on Research and Development in Information Retrieval, SIGIR 2020*. ACM, 1997–2000.

[8] Maik Fröbe, Jan Philipp Bittner, Martin Potthast, and Matthias Hagen. The Effect of Content-Equivalent Near-Duplicates on the Evaluation of Search Engines. In *Proceedings of the 42nd European Conference on IR Research, ECIR 2020*. Springer, 12–19.

[9] Lukas Gienapp, Maik Fröbe, Matthias Hagen, and Martin Potthast. Sparse Pairwise Re-ranking with Pre-trained Transformers. In *Proceedings of the 2022 ACM SIGIR International Conference on the Theory of Information Retrieval, ICTIR 2022*. ACM, 72–80.

[10] Matthias Hagen, Michael Völske, Steve Göring, and Benno Stein. Axiomatic Result Re-Ranking. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016*. ACM, 721–730.

[11] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. PubMedQA: A Dataset for Biomedical Research Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*. ACL, 2567–2577.

[12] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. UnifiedQA: Crossing Format Boundaries With a Single QA System. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. ACL, 1896–1907.

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019).

[14] Xueguang Ma, Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. Document Expansion Baselines and Learned Sparse Lexical Representations for MS MARCO V1 and V2. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2022*. ACM, 3187–3197.

[15] Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. Simplified Data Wrangling with ir_datasets. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2021*. ACM, 2429–2436.

[16] Craig Macdonald, Nicola Tonellotto, Sean MacAvaney, and Iadh Ounis. PyTerrier: Declarative Experimentation in Python from BM25 to Dense Retrieval. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management, CIKM 2021*. ACM, 4526–4533.

[17] Rodrigo Nogueira and Jimmy Lin. From doc2query to docTTTTTquery. *Online preprint* (2019).

[18] 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, SIGIR 2021*. ACM, 2066–2070.

[19] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. *CoRR* abs/2101.05667 (2021).

[20] 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. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.

[21] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994*. NIST, 109–126.

[22] 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, NAACL-HLT 2018*. ACL, 809–819.

[23] David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. LongChecker: Improving Scientific Claim Verification by Modeling Full-Abstract Context. *CoRR* abs/2112.01640 (2021).

[24] David Wadden, Kyle Lo, Lucy Lu 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*. ACL, 61–76.

[25] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. LinkBERT: Pretraining Language Models with Document Links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*. ACL, 8003–8016.

[26] 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 2022*. ACM, 2099–2104.