Webis at TREC 2021: Deep Learning, Health Misinformation, and Podcasts Tracks

Alexander Bondarenko [1,*] Maik Fröbe [1,*] Marcel Gohsen [2,*] Sebastian Günther [1,*]
Johannes Kiesel [2,*] Jakob Schwerter [3,*] Shahbaz Syed [3,*] Michael Völske [2,*]
Martin Potthast [3] Benno Stein [2] Matthias Hagen [1]

[1]Martin-Luther-Universität Halle-Wittenberg ⟨first⟩.⟨last⟩@informatik.uni-halle.de
[2]Bauhaus-Universität Weimar ⟨first⟩.⟨last⟩@uni-weimar.de
[3]Leipzig University ⟨first⟩.⟨last⟩@uni-leipzig.de

ABSTRACT

We describe the Webis group's participation in the TREC 2021 Deep Learning, Health Misinformation, and Podcasts tracks. Our three LambdaMART-based runs submitted to the Deep Learning track focus on the question whether anchor text is an effective retrieval feature in the MS MARCO scenario. In the Health Misinformation track, we axiomatically re-ranked the top-20 results of BM25 and MonoT5 for argumentative topics. As for the Podcasts track, our submitted six runs focus on supervised classification of podcasts as entertaining, subjective, or containing discussion by using audio and text embeddings.

1 INTRODUCTION

We have participated in three TREC 2021 tracks: Deep Learning, Health Misinformation, and Podcasts. With our three runs submitted to the Deep Learning track, we investigate the effectiveness of anchor texts in the current MS MARCO setup inspired by the two decades old statement that anchors "often provide more accurate descriptions of web pages than the pages themselves" [6]. Therefore, we use the Webis MS MARCO Anchor Text 2022 dataset [13] consisting of anchor text pointing to MS MARCO documents and combine anchor text with other features in LambdaMART models.

Our runs for the Health Misinformation track target topics that seem to be of argumentative nature, and we then attempt to rank documents higher that contain "good" argumentation. The underlying assumption is that well-argued texts contain fewer wrong, unreliable, or misleading information. Respective results for queries like "Should I apply ice to a burn?" then might not lead searchers to wrong decisions that negatively affect their health.

In the Podcasts Track, for retrieval, we classify episodes as entertaining, subjective, or containing discussions by using models that we had trained on a small set of manually labeled episodes. We test re-rankings of standard result lists by using the classifiers' confidences based on audio embeddings, text embeddings, or both combined. To generate engaging summaries, we employ only sentences classified as entertaining by our combined model. We investigate both extractive and abstractive summarization.

*These authors contributed to the paper equally.

TREC 2021, November 15–19, 2021, Gaithersburg, Maryland

2 DEEP LEARNING TRACK

Transformer-based re-rankers caused a paradigm shift in information retrieval [19]—also evident in the Deep Learning tracks with transformer-based approaches being the most effective in the previous years [10, 11]. We submitted three traditional learning-to-rank runs to the TREC 2021 Deep Learning track to help to diversify the judgment pool and to allow the comparison of modern transformer-based rankings with more traditional baselines. Within our three runs, we focus on the research question of whether anchor text is an effective feature in the MS MARCO scenario. We train two LambdaMART [7] models that use anchor texts as feature type (among others) and one LambdaMART model without anchor text.

Table 1 provides an overview of the 50 feature types that we use in the learning-to-rank models. To calculate the query–document similarity, we use four types of text: (1) the document body, (2) the document title, (3) the URL, and (4) anchor text pointing to the document. For each of these four text types, we calculate nine feature types using Anserini [31]: BM25, F2Exp, F2Log, PL2, QL, QLJM, SPL, TF, and TF · IDF scores. In addition, we use eight query-independent document feature types including the Alexa Rank[1] or Harmonic Mean and PageRank of the domain provided by the Common Crawl,[2] as well as six query features since such features can be used by LambdaMART to learn different subtrees for different types of queries [22].

As our anchor text source, we use the Webis MS MARCO Anchor Text 2022 dataset [13].[3] This dataset enriches 1,703,834 documents for Version 1 and 4,821,244 documents for Version 2 of the MS MARCO document collection with up to 1000 anchor texts extracted from six Common Crawl snapshots from the years 2016 to 2021 (1.7–3.4 billion documents). The anchor text dataset already is filtered to remove "low quality" anchor texts (intra-site anchor texts, very long anchor texts that are very long, and stopword-only anchor texts). Here, we use only anchor texts extracted from the Common Crawl snapshot 2021-04 since it seems to be the snapshot in the Webis MS MARCO Anchor Text 2022 dataset closest to the crawling time of Version 2 of MS MARCO. We thus use 60.62 million anchor texts pointing to 1.14 million documents.

Using the official training and validation labels for Version 2 of the MS MARCO document collection, we train LambdaMART

[1]web.archive.org/web/202104010000/s3.amazonaws.com/alexa-static/top-1m.csv.zip
[2]commoncrawl.org/2021/05/host-and-domain-level-web-graphs-feb-apr-may-2021/
[3]Data available at https://github.com/webis-de/ECIR-22 or https://zenodo.org/record/5883456 but also integrated in `ir_datasets` [21].

Table 1: (Left) Feature types employed in our Deep Learning track runs (9 anchor text-based query–document feature types not used in run webis-dl-2). (Right) Most important feature types of the LambdaMART model with 5,000 trees in our pilot study.

Query–Document		Document/Domain		Query	
Description	Count	Description	Count	Description	Count
BM25 score	4	Alexa Rank	1	Is 5W1H question	1
F2Exp score	4	Body length	1	Geopol. entities	1
F2Log score	4	Dots in host	1	Is comparative [3]	1
PL2 score	4	Harmonic Mean	1	Length in tokens	1
QL score	4	PageRank	1	ORG entities	1
QLJM score	4	Slashes in URL	1	Person entities	1
SPL score	4	Title length	1		
TF score	4	URL length	1		
TF · IDF score	4				
Total: 50 feature types, 9 query–document ones are anchor text-based					

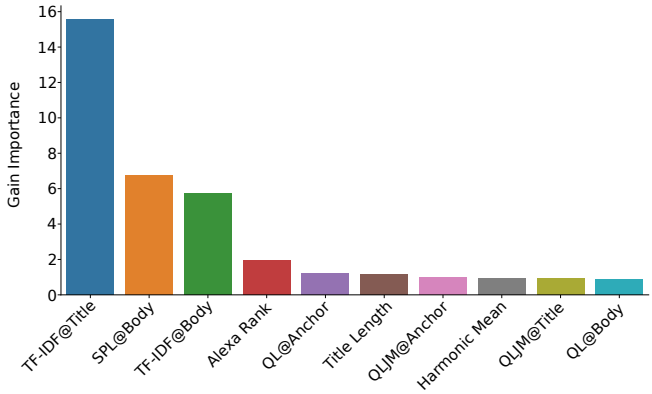

Table 2: The Deep Learning track's official effectiveness results for our three runs in comparison to the official baseline—our runs re-rank this baseline's top-100 results.

	nDCG@3	nDCG@10	P@1	P@3	P@10	MRR@10
webis-dl-1	**0.63**	0.58	0.91	0.85	0.74	0.94
webis-dl-2	0.61	0.57	0.91	0.84	0.73	0.94
webis-dl-3	**0.63**	**0.59**	**0.93**	**0.86**	**0.75**	**0.95**
baseline	0.54	0.51	0.75	0.74	0.67	0.84

models—a feature-based learning to rank approach [7, 16, 30]. In a pilot study, we trained LambdaMART models with 100, 1,000, and 5,000 trees optimized for MRR using the LightGBM framework [17] on all 50 feature types; other hyperparameters at their default values. The model with 5,000 trees had a higher MRR on the validation set than the models with 1,000 or 100 trees. The right plot in Table 1 shows the 10 most important feature types for the model with 5,000 trees: TF · IDF-scored titles are the most important but QL- and QLJM-scored anchor texts follow at positions 5 and 7.

To also assess the effectiveness of anchor text in the TREC 2021 Deep Learning track setup, we submitted the following three runs.

webis-dl-1. LambdaMART model with 5,000 trees (other hyperparameters at their default values) trained on all 50 feature types (cf. Table 1); re-ranks the top-100 results of the track's official baseline.

webis-dl-2. LambdaMART model with 5,000 trees (other hyperparameters at their default values) trained on the 41 feature types without the anchor text-based feature types; re-ranks the top-100 results of the track's official baseline.

webis-dl-3. LambdaMART model with 1,000 trees (fewer parameters to prevent overfitting on the training data; other hyperparameters at their default values) trained on all 50 feature types (cf. Table 1); re-ranks the top-100 results of the track's official baseline.

Table 2 shows the official effectiveness results of our runs and of the official baseline. All our runs improve upon the baseline but none of the improvements are statistically significant at a p-value of 0.05 with Bonferroni correction. Our webis-dl-3 run (1,000 trees) achieves the highest effectiveness, indicating that using 5,000 trees might overfit to the training data (even though 5,000 trees obtained higher MRR scores on the validation set in our pilot study). Comparing webis-dl-1 (with anchor text) to webis-dl-2 (without anchor text) shows that anchor text can very slightly improve the effectiveness. Still, since anchor text is assumed to be most effective for navigational queries [9, 13] and since none of the TREC 2021 Deep Learning track's topics are intended to be navigational, probably no larger effect was to be expected.

3 HEALTH MISINFORMATION TRACK

Our six runs for the Health Misinformation track focus on investigating whether re-ranking with axioms that capture facets of argumentativeness can improve the "helpfulness" while reducing the "harmfulness" of results for queries that seem to be of argumentative nature (i.e., not just factual look-ups).

3.1 Initial Retrieval with BM25 and MonoT5

We index the document collection using Anserini [31] (plain text extraction with Jsoup, stopword removal with Lucene's default stopword list for English, Porter stemming). Our first baseline run webis-bm25 uses Anserini's BM25 implementation (default values $k = 0.9$, $b = 0.4$), our second baseline run webis-t5 uses PyGaggle's[4] default MonoT5 model [26] castorini/monot5-base-msmarco to re-rank the top-50 BM25 results.

3.2 Argumentative Axiomatic Re-ranking

For each of the two baselines (BM25 and MonoT5), we have two further runs that re-rank the baseline's top-20 results for queries that seem to be argumentative (i.e., not just "simple" factual look-ups). The re-ranking is based on axioms capturing argumentativeness and we largely applying the same re-ranking strategy as in our previous TREC participations [2, 4, 5].

3.2.1 Identifying Argumentative Queries. Based on the assumption that users submitting "argumentative" queries (i.e., not simple factual look-ups) will prefer documents containing good argumentation—that then might contain fewer wrong or misleading

[4] https://github.com/castorini/pygaggle

Table 3: The Health Misinformation track's official effectiveness results for our runs. U: useful, Co: correct, Cr: credible, Incor.: incorrect.

Run	Compatibility		nDCG (binary)			P@10 (binary)		nDCG (graded)
	Help	Harm	U & Co	U & Cr	U & Co & Cr	U & Co	Incor.	Useful
webis-bm25	**0.13**	**0.14**	**0.43**	**0.49**	0.38	0.31	0.29	**0.58**
webis-bm25-ax1	**0.13**	**0.14**	**0.43**	**0.49**	**0.39**	0.31	**0.28**	**0.58**
webis-bm25-ax3	**0.13**	**0.14**	**0.43**	**0.49**	0.38	0.31	**0.28**	**0.58**
webis-t5	**0.13**	**0.14**	0.24	0.26	0.19	0.32	0.30	0.34
webis-t5-ax1	**0.13**	**0.14**	0.24	0.27	0.19	**0.35**	0.33	0.34
webis-t5-ax3	**0.13**	**0.14**	0.24	0.26	0.19	0.34	0.33	0.34

information—, we first check which of the Health Misinformation track's topics are argumentative. We manually inspected all topics and, given their non-factual health-related nature (e.g., "Should I apply ice to a burn?"), concluded that all of them could be labeled as argumentative queries.

3.2.2 Re-ranking Axioms. To re-rank the top-20 baseline results, we use three axioms that prefer argumentative documents from our previous years' runs submitted to the Common Core, Decision, and Health Misinformation tracks [2, 4, 5].

Retrieval axioms (i.e., formally defined constraints applied to retrieval models) have been developed within the field of axiomatic thinking in information retrieval [1] to define heuristic constraints that good retrieval models should probably fulfill. A basic example is the term frequency axiom TFC1 [12]: for a single-term query, from two documents of the same length, the document with more query term occurrences should receive a higher retrieval score. We follow the ideas of axiomatic thinking and apply three axioms that capture document argumentativeness. In contrast to the restrictive "same length"-assumption in TFC1, we relax the preconditions to ensure the axioms' practical applicability.

Axiom ArgUC (Argumentative Units Count). The general idea of the ArgUC axiom is to favor documents that contain a larger number of argumentative units.

Formalization. Let Q be an argumentative query, D_1 and D_2 be two documents retrieved for Q, let $count_{\text{Arg}}(\cdot)$ be the number of argumentative units in a document, and let $\approx_{10\%}$ indicate "equality" up to a 10% difference. If $length(D_1) \approx_{10\%} length(D_2)$ and $count_{\text{Arg}}(D_1) > count_{\text{Arg}}(D_2)$, then D_1 should be ranked higher than D_2.

Axiom QTArg (Query Term Occurrence in Argumentative Units). Retrieved documents usually consist of argumentative and non-argumentative units or text passages. The general idea of the QTArg axiom is to favor documents where the query terms appear closer to or in more argumentative units.

Formalization. Let $Q = \{q\}$ be an argumentative single-term query, D_1 and D_2 be two retrieved documents, and let Arg_D be the set of argumentative units of a document D. If $length(D_1) \approx_{10\%} length(D_2)$ and $q \in A_{D_1}$ for some $A_{D_1} \in Arg_{D_1}$ but $q \notin A_{D_2}$ for all $A_{D_2} \in Arg_{D_2}$, then D_1 should be ranked higher than D_2.

Axiom QTPArg (Query Term Position in Argumentative Units). Following the general observation that in relevant documents the first occurrences of query terms are closer to the beginning of the document [24, 29], the QTPArg axiom favors documents where the first appearance of a query term in an argumentative unit is closer to the beginning of the document.

Formalization. Let $Q = \{q\}$ be an argumentative single-term query, D_1 and D_2 be two retrieved documents, and let the first position in an argumentative unit of a document D where the term q appears be denoted by $1^{st}position(q, Arg_D)$. If $length(D_1) \approx_{10\%} length(D_2)$ and $1^{st}position(q, Arg_{D_1}) < 1^{st}position(q, Arg_{D_2})$, then D_1 should be ranked higher than D_2.

3.2.3 Argumentative Unit Detection. The above argumentative axioms are based on argumentative units. To detect argumentative units in documents, we use the BiLSTM-CNN-CRF-based argument tagging tool `TARGER` [8]; available via an API.[5] As input, `TARGER` accepts a text, and it returns markers indicating the beginning and end of argument premises and claims (argumentative units).

3.2.4 Actual Runs. In addition to the three argumentative axioms, we also employ the axiom ORIG [14] that simply returns the preferences of the initial BM25 or MonoT5 ranking. We do this to "balance" argumentativeness and topical relevance. Differing from the the original axiomatic re-ranking pipeline [14], we do not train the weights for the axioms but apply two simple strategies distributing a total weight of 1.0 to the four axioms: (a) we manually set the weights such that at least two argumentative axioms have to agree to "overrule" the ORIG preference (runs webis-bm25-ax1 and webis-t5-ax1; weights: ORIG gets 0.22 and the argumentative axioms each get 0.26) or (b) we manually set the weights such that all three argumentative axioms have to agree to overrule the ORIG preference (runs webis-bm25-ax3 and webis-t5-ax3; weights: ORIG gets 0.49 and the argumentative axioms each get 0.17).

3.3 Evaluation

In the Health Misinformation track, the retrieval effectiveness of the submitted runs is evaluated using nDCG, precision, and a compatibility measure. To this end, NIST assessors labeled the usefulness, correctness, and credibility of documents. The compatibility measure combines these three aspects into preference orderings for the document helpfulness and harmfulness.

The results for our runs in Table 3 show that no real effect of the axiomatic re-ranking can be observed. The reason is that the axioms very rarely overrule the ORIG preference.

[5] API: https://demo.webis.de/targer-api/apidocs/; Python library: https://pypi.org/project/targer-api/

Table 4: F1 scores for our Podcast track runs' SVM classification of segments as entertaining, subjective, or containing a discussion; 10-fold cross validation on our annotated 1,000 podcast segments.

Run	Entertaining	Subjective	Discussion
Webis_pc_cola	0.54	0.69	0.71
Webis_pc_rob	0.58	**0.78**	0.77
Webis_pc_cola_rob	**0.63**	0.75	**0.80**

4 PODCASTS TRACK

We participated in the retrieval and summarization tasks of the TREC 2021 Podcasts track.

4.1 Retrieval Task

We submitted four runs for this task employing Elasticsearch's BM25 retrieval model. These four runs consist of one baseline without re-ranking and three runs including a supervised re-ranking method that use audio features, text features, and both, respectively.

For the supervised methods, we manually annotated 1,000 podcast segments (length of 2 minutes) for each of the three re-ranking criteria: entertaining, subjective, discussion. Each podcast segment was annotated by a single annotator. We used the training queries and our baseline to retrieve these segments. Using an SVM classifier, the different feature sets are then employed to train (on the annotated data) and predict (on the segments retrieved for each topic in the final evaluation) whether a segment is entertaining, subjective, or contains a discussion. The results are then re-ranked by multiplying their BM25 score with the SVM's confidence for the respective criterion (a value between 0 and 1).

4.1.1 Actual Runs. The run Webis_pc_bs retrieves the top-1,000 BM25 results and serves as the baseline that the other runs re-rank. The SVM for the run Webis_pc_cola employs audio embeddings extracted from the podcast audio clips using a COLA model [28]. We trained the embedding model on 10,000 hours of randomly selected podcast episodes from the corpus. The SVM for the run Webis_pc_rob employs text embeddings from the podcast transcripts using an out-of-the-box RoBERTa model [20]. The SVM for the run Webis_pc_co_rob employs the audio and the text embeddings.

4.1.2 Evaluation. Table 4 shows the F1 scores of the classifiers used in the non-baseline runs with respect to the three ranking criteria (entertaining, subjective, discussion) in a 10-fold cross validation on our annotated sample of 1,000 podcast segments. Overall, the effectiveness of the classifier using audio and text features is best on the entertaining and discussion criteria, while employing text features only is the most effective for the subjective criterion.

Table 5 shows the effectiveness of our runs for each ranking criterion. Each re-ranking method actually reduced the performance according to all measures. Further investigations are needed to identify the reasons for this consistent drop. Possible explanations are a misconception of the ranking criteria on our side (leading to false annotations of the training data) or general problems in the generalizability of our approach.

Table 5: The Podcast track's official effectiveness results for our retrieval task runs and each ranking criterion.

Criterion	Run	nDCG@30	nDCG@1000	P@10
Entertaining	Webis_pc_bs	**0.12**	**0.23**	**0.10**
	Webis_pc_cola	0.05	0.17	0.05
	Webis_pc_rob	0.04	0.16	0.03
	Webis_pc_co_rob	0.03	0.16	0.03
Subjective	Webis_pc_bs	**0.17**	**0.34**	**0.20**
	Webis_pc_cola	0.06	0.24	0.06
	Webis_pc_rob	0.04	0.23	0.04
	Webis_pc_co_rob	0.04	0.23	0.06
Discussion	Webis_pc_bs	**0.16**	**0.32**	**0.16**
	Webis_pc_cola	0.06	0.23	0.06
	Webis_pc_rob	0.04	0.21	0.04
	Webis_pc_co_rob	0.05	0.22	0.06

4.2 Summarization Task

We submitted two runs for this task, an abstractive and an extractive summarization system. We tested to generate summaries biased towards the most entertaining sentences of the episodes, which we expect to be especially suitable to encourage consumers to listen to the whole episode. We employ the combined features model of the retrieval task (Section 4.1) to determine which sentences are entertaining, and use the confidence of the classifier as a proxy for how entertaining a sentence is.

4.2.1 Actual Runs. The run Webis_pc_abstr (abstractive) employs a DistilBART[6] summarization model [18], finetuned on the CNN/DailyMail corpus [15, 25] to create a textual summary for an episode. As input to the model, we use a concatenation of up to 25 sentences in their original relative ordering in the episode: the 5 sentences with the highest confidence for being entertaining according to our SVM classifier, as well as the respective previous and following two sentences (when they exist). The audio summary of this run is not abstractive but simply concatenates the up to 25 selected sentences in their original relative ordering in the episode until a length of one minute is reached or all 25 sentences are included.

The run Webis_pc_extr (extractive) employs the TextRank algorithm [23] to rank the episode's sentences, using the 10 highest ranked sentences as the summary—in their original relative ordering in the episode. TextRank applies PageRank on a weighted graph with nodes representing sentences and the starting edge weights being the "similarity" of the sentences. As the similarity, we employed a sum of two components: (1) the cosine similarity of their sentence embeddings, extracted using a sentence transformer model[7] [27], and (2) the sum of our SVM classifier's entertainment confidences for the two sentences. The audio summary concatenates the 10 sentences until a length of one minute is reached or all 10 sentences are included.

4.2.2 Evaluation. Table 6 shows the effectiveness of both runs. Without achieving any "excellent" score and only very few "good" scores, both runs perform very poorly—substantially below the one-minute baseline. Further investigations are needed to identify

[6] https://huggingface.co/sshleifer/distilbart-cnn-12-6

[7] https://huggingface.co/sentence-transformers/paraphrase-xlm-r-multilingual-v1

Table 6: The Podcast track's official effectiveness results for our summarization task runs on 193 episodes. The EGFB score is the average of all episodes, with E(xcellent) as 4, G(ood) as 2, F(air) as 1, and B(ad) as 0.

Run	EGFB score	E	G	F	B
Webis_pc_abstr	0.23	0	6	33	154
Webis_pc_extr	0.26	0	6	38	148
Baseline (one-minute)	**0.81**	7	26	76	84

reasons for this poor summarization quality. Like for retrieval, possible explanations are a misconception of the ranking criteria on our side (leading to false annotations of the training data) or general problems in the generalizability of our approaches. Another problem might be that our hypothesis is wrong (focusing on entertaining sentences leads to especially engaging summaries) and would have led to a poor quality even if the classification had worked perfectly.

5 CONCLUSION

In the Deep Learning track, we investigated the effectiveness of anchor text-based features in LambdaMART-based re-rankings of the official track baseline. The experimental evaluation only showed some very slight improvements when anchor text-based features are used—the non-navigational topics of the TREC 2021 Deep Learning track simply do not really benefit from anchor text.

In the Health Misinformation track, we investigated whether axiom-induced preferences that capture argumentativeness can lead to more relevant and helpful search results. The experimental evaluation showed no effect since the axioms hardly ever changed the BM25 or MonoT5 baseline ranking preferences.

In the Podcasts track, we investigated the effectiveness of supervised approaches using both text and audio embeddings. The experimental evaluation showed that our re-ranking decreased the effectiveness for every instance in the retrieval task while our summaries generated from entertaining sentences could not reach the one-minute baseline in the summarization task.

ACKNOWLEDGMENTS

This work has been partially supported by the DFG through the project "ACQuA: Answering Comparative Questions with Arguments" (grants HA 5851/2-1 and HA 5851/2-2) as part of the priority program "RATIO: Robust Argumentation Machines" (SPP 1999). Some computations for this work were done using resources of the Leipzig University Computing Center.

REFERENCES

[1] Enrique Amigó, Hui Fang, Stefano Mizzaro, and ChengXiang Zhai. Axiomatic Thinking for Information Retrieval: And Related Tasks. In *Proceedings of the 40th International Conference on Research and Development in Information Retrieval (SIGIR 2017)*. ACM, 1419–1420.

[2] Janek Bevendorff, Alexander Bondarenko, Maik Fröbe, Sebastian Günther, Michael Völske, Benno Stein, and Matthias Hagen. Webis at TREC 2020: Health Misinformation Track. In *Proceedings of the 29th International Text Retrieval Conference (TREC 2020)*. NIST, 4.

[3] Alexander Bondarenko, Pavel Braslavski, Michael Völske, Rami Aly, Maik Fröbe, Alexander Panchenko, Chris Biemann, Benno Stein, and Matthias Hagen. Comparative Web Search Questions. In *Proceedings of the 13th ACM International Conference on Web Search and Data Mining (WSDM 2020)*. ACM, 52–60.

[4] Alexander Bondarenko, Maik Fröbe, Vaibhav Kasturia, Michael Völske, Benno Stein, and Matthias Hagen. Webis at TREC 2019: Decision Track. In *Proceedings of the 28th International Text Retrieval Conference (TREC 2019)*. NIST, 4.

[5] Alexander Bondarenko, Michael Völske, Alexander Panchenko, Chris Biemann, Benno Stein, and Matthias Hagen. 2018. Webis at TREC 2018: Common Core Track. In *27th International Text Retrieval Conference (TREC 2018) (NIST Special Publication)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST), 3.

[6] Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks* 30, 1-7 (1998), 107–117.

[7] Christopher JC Burges. 2010. From RankNet to LambdaRank to LambdaMART: An Overview. *Learning* 11, 23-581 (2010), 81.

[8] Artem Chernodub, Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. TARGER: Neural Argument Mining at Your Fingertips. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*. ACL, 195–200.

[9] Nick Craswell, David Hawking, and Stephen E. Robertson. Effective Site Finding Using Link Anchor Information. In *Proceedings of the 24th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2001)*. ACM, 250–257.

[10] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. Overview of the TREC 2020 Deep Learning Track. In *Proceedings of the 29th Text REtrieval Conference (TREC 2020)* . NIST, 13.

[11] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. Overview of the TREC 2019 Deep Learning Track. *CoRR* abs/2003.07820 (2020).

[12] Hui Fang, Tao Tao, and ChengXiang Zhai. A Formal Study of Information Retrieval Heuristics. In *Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2004)*. ACM, 49–56.

[13] Maik Fröbe, Sebastian Günther, Maximilian Probst, Martin Potthast, and Matthias Hagen. The Power of Anchor Text in the Neural Retrieval Era. In *Proceedings of the 44th European Conference on IR Research (ECIR 2022)*. Springer, Berlin Heidelberg New York.

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

[15] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching Machines to Read and Comprehend. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS 2015)*. 1693–1701.

[16] Ziniu Hu, Yang Wang, Qu Peng, and Hang Li. Unbiased LambdaMART: An Unbiased Pairwise Learning-to-Rank Algorithm. In *Proceedings of the World Wide Web Conference (WWW 2019)*. ACM, 2830–2836.

[17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS 2017)*. 3146–3154.

[18] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. ACL, 7871–7880.

[19] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained Transformers for Text Ranking: BERT and Beyond. *Synthesis Lectures on Human Language Technologies* 14, 4 (2021), 1–325.

[20] 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).

[21] 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 Conference on Research and Development in Information Retrieval (SIGIR 2021)*. ACM, 2429–2436.

[22] Craig Macdonald, Rodrygo L. T. Santos, and Iadh Ounis. On the Usefulness of Query Features for Learning to Rank. In *Proceedings of the 21st International Conference on Information and Knowledge Management (CIKM 2012)*. ACM, 2559–2562.

[23] R. Mihalcea and P. Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*. ACL, 404–411.

[24] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to Match Using Local and Distributed Representations of Text for Web Search. In *Proceedings of the 26th International Conference on World Wide Web (WWW 2017)*. ACM, 1291–1299.

[25] Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of the 20th Conference on Computational Natural Language Learning (CoNLL 2016)*. ACL, 280–290.

[26] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2020) (Findings of ACL)*. ACL, 708–718.

[27] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*. ACL, 3980–3990.

[28] Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive Learning of General-Purpose Audio Representations. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021)*. IEEE, 3875–3879.

[29] Adam D. Troy and Guo-Qiang Zhang. Enhancing Relevance Scoring with Chronological Term Rank. In *Proceedings of the 30th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2007)*. ACM, 599–606.

[30] Qiang Wu, Christopher J. C. Burges, Krysta M. Svore, and Jianfeng Gao. Adapting Boosting for Information Retrieval Measures. *Inf. Retr.* 13, 3 (2010), 254–270.

[31] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *Proceedings of the 40th International Conference on Research and Development in Information Retrieval (SIGIR 2017)*. ACM, 1253–1256.