TKB48 at TREC 2021 News Track

Zhang Lirong
Graduate School of Comprehensive Human Sciences, University of Tsukuba
Tsukuba, Ibaraki, Japan
s2021710@s.tsukuba.ac.jp

Hideo Joho
Faculty of Library, Information and Media Science, University of Tsukuba
Tsukuba, Ibaraki, Japan
hideo@slis.tsukuba.ac.jp

Sumio Fujita
Yahoo Japan Corporation
Tokyo, Japan
sufujita@yahoo-corp.jp

ABSTRACT

TKB48 incorporated document expansion methods such as docT5query and keyword extraction into indexing to solve the background linking problem. Using a transformer-based model, we calculated the text similarity of queries and documents at a semantic level and combined the semantic similarity and BM25 score for re-ranking background articles. We examined different combinations of re-ranking factors such as semantic similarities between expanded documents and attributes of topics. We found that increasing index fields produced by the docT5query model and keyword extraction model was beneficial. At the same time, the re-ranking performance was influenced by the amount of semantic similarity factors and their weight in the total relevance score. To discover the effectiveness of document expansion and our method using temporal recency, we further generated several unofficial runs incorporating a temporal topic classifier and learning to rank method. However, the lack of temporal topics limits the performance of the model. Our purposed algorithm outperformed the learning to rank method. Our future work will focus on fine-tuning of the docT5query model.

KEYWORDS

Document Expansion, Temporal Recency, Information Retrieval

1 INTRODUCTION

According to Pew Research, in 2018, 93% of American adults consume at least some of the news online, while the number was 38% in 2016, which shows that the percentage of people consuming news on the web is rapidly increasing [15]. Usually for people reading a news story, they need to get reference information or background knowledge from other articles. Therefore, developing a method to efficiently locate the background knowledge needed to understand an article is very relevant to the current retrieval needs of people. Common applications include providing reference links alongside news articles to help users access the background knowledge they may need more efficiently Or recommending the next article the user should read. Motivated by the above, the News Track of TREC sets two sub-tasks: Background Linking and Wikification this year. Background Linking and Wikification. Background Linking emphasizes providing a recommended articles list containing background knowledge or contextual information that helps users understand the complete news story. Wikification is the automatic hyperlinking of entities, concepts, or references to another resource that provides more information on the linked thing[1]. This year, we participated in the Background Linking task.

[1]http://trec-news.org/guidelines-2021.pdf

The new feature of this year's Background Linking is a new element called subtopics which represent reasons for seeking background. Participants need to ensure that the retrieved articles meet these reasons and also ensure the diversity of the results. In this paper, we investigated the capability of docT5query model and transformer-based re-ranking methods in the background linking task. We also adopted a method using temporal features of topics and documents.

2 RELATED WORK

Most common methods in background linking are keyword extraction or named entity recognition combining with query expansion and ad-hoc search [2, 10, 12]. Lu and Fang [11] proposed a new way that extracted aspects from the constructed graph relations based on the entities, then generated the final ranking scores based on the likelihood of aspect and article language model. Ornella and Gianmaria [7] also proposed entity graph methods. Differently, they considered document feature vectors as a combination of textual and graph-based features and applied them in a learning to rank model.

Methods based on transformer models are also commonly used. SU-NLP 2020 [1] proposed to use BERT summarization model to extract useful paragraphs from long articles. They indexed documents with vectors mapped by a sentence encoder and performed retrieval on the cosine similarity between query and doc. Another work [5] performed ranking according to the semantic similarity which calculated by sentence-BERT between documents and queries. Based on the assumption that similar articles have similar embedding vectors, ClaC Lab 2020 [9] leveraged a variety of embedding models, including BERT and retrieved background articles on their similarity scores.

BM25-based searching is a common baseline in News Track. However, OSC 2020 [4] used more like this function of Elasticsearch as their baseline searching method. MLT(more like this) can extract import terms from a query doc and conduct BM25-based searching using queries composed of these terms. In this work, we continue to use BM25 as our base retrieval method.

There was also a potential problem with background article retrieval tasks, where they typically used the article itself to retrieve articles. This results in the content of the retrieved articles overlapping with articles that the user had already read and did not serve to provide background knowledge. Therefore we proposed a query prediction model and used the output to extend the original article. This allows the origin contents include queries will be issued for a given document. This adds diversity to the background article search.

DocT5query model is derived from doc2query. Doc2query was first introduced as a document expansion method [14] and achieved the best run on TREC CAR [6]. The doc2query model can be trained as a sequence-to-sequence model with the dataset of query and relevant document pairs. Recently, a research [13] shows training doc2query model as a transformer T5 model is more efficient and can get higher accuracy than a sequence-to-sequence transformer model[16]. Thus we will adopt docT5query as our prediction model.

3 METHOD

3.1 Dataset and Preprocess

This year's dataset is the fourth version of TREC Washington Post Collection, which contains 728,626 news articles and blog posts from January 2012 through December 2020. We re-formatted it with the attributes of id, title, date of publication, content, and source and de-duplicated the corpus with id, title, and publication date.

3.2 Document Expansion

For a given article, we inferred the queries that may lead to that article. We believe that using the queries inferred from the article can help retrieve relevant articles that match such queries. We made query predictions for each article in the corpus and add them as a new field to be indexed. If the contents of two articles are related to each other, then their predicted queries can be related to a near event or topic. We assume that a search conducted within the target document's predicted queries has a great probability, leading to articles carrying contextual information. We perform our query prediction using docT5query[2] model.

DocT5query model adopts the T5Tokenizer pre-trained from MS Marco dataset. We cleaned the text by removing the image hyperlinks and URLs before input into docT5query model. We set the model to output the top 10 most possible queries for this article.

We hypothesized that the keyword extraction method was able to pull out words that represent the central content of the article. These words are also very likely to contain the names of events contained in the article. Adding the extracted keyword to the index allows the article to gain higher relevance scores when retrieved with related queries. Keyword extractions are performed by the external tool called PKE[3].

We adopted an unsupervised graph-based model called MultipartiteRank [3] as our keyword extraction method. The model will build the multipartite graph based on the longest sequences after removing punctuation and stop words. We used the default candidate weight setting and set the model to output ten best keywords.

3.3 Semantic Similarity

Predicted queries may be distribute across varied aspects. For instance, an article talking about an author's death may refer to queries like what awards he/she got, what books he/she wrote, and what one of his/her books is talking about. Using the semantic similarity between two articles can help us determine whether the retrieved result-articles content fits with the content of the query article or not. We assumed that articles with higher semantic similarity are more likely to provide more valuable contextual content. For example, two articles discuss the same event from different perspectives, or two articles talk about the impact of the same event in various fields. We use the sentence-transformer to embed sentence vectors and adopt dot products as the semantic similarity score. We leverage articles' semantic relevance from three aspects: between the article content and the topic description, between the keywords of the article and the topic description, and between the article keywords and the topic description's keywords.

Table 1: Runs Explanation

runs	retrieve condition		
	query components	***re-ranking factors***	***index fields***
TKB48_Run1	Topics' desc Predicted Queries	S_{DK} S_{DC}	Title, Content Predicted queries Key words
TKB48_Run2	Same as above	+ W_R	Same as above
TKB48_Run3	Same as above	+ S_{KK}	Same as above
TKB48_Run4	+ Topics' title	Same as above	Same as above

3.4 Temporal Recency

Many of the teams [1, 11] in previous years added a temporal filter to the search process, assuming that the article which would provide context for the query article must have been posted before it. Based on such assumption, they filtered out any articles whose time came after the query article. However, we believe that articles posted after the query article may also provide background information. For example, articles that follow an event and talk about its effects and consequences are often published after the target article. To be more specific, for example, articles talking about the COVID-19 vaccination are probably posted after articles talking about the COVID-19 outbreak, moreover they provide contextual information to each other. Therefore, we did not set a filter this time but used the time freshness between such two articles to increase the ranking of articles in a near time period.

$$W_R = \frac{y}{t_d + y} \quad t_d \in R \tag{1}$$

Temporal recency is calculated as equation 1 where y is in the constant of 365, t_d is the day distance between two articles.

3.5 Submitted Runs

This year we submitted four runs for each Background Linking and Background Linking (subtopics) task.

3.5.1 Background Linking Runs. The table 1 shows the search conditions between different runs, each with a new part added to the previous run.

"Same as above" means the condition did not change referring to the previous run. S_{DC}, S_{DK}, and S_{KK} represent the semantic similarity from three different aspects: the topic desc with article content, the topic description with the keywords of the article, and the topic description's keywords with the article keywords.

[2]https://github.com/castorini/docTTTTTquery
[3]https://github.com/BluceHan/pke

Table 2: Subtopic Runs Explanation

runs	retrieve condition		
	query components	*re-ranking factors*	*index fields*
TKB48_SRun1	Topics' desc Subtopics	S_{DC} S_{DK} S_{KK}	Title, Content Predicted queries Key words
TKB48_SRun2	Same as above	+ W_R	Same as above
TKB48_SRun3	+ Predicted Queries	Same as above	Same as above
TKB48_SRun4	Same as above	$+S_{STC}$	Same as above

Algorithm 1 RetrieveByPredictedQuery

```
querydoc ← required query document
sq ← search query put into Solr
QL ← DocT5model(querydoc)
for query in QL do
    sq = topic's title + desc + query
    res_i = SolrSearchTop100(sq)
end for
res ← Sort&Deduplicate(res_i)
```

We conducted the indexing process and BM25 retrieval process by Apache Solr[4]. For Baseline 1 to 4, we retrieved the top 200 documents. In Baselines 3 and 4, we calculated the semantic similarity score and re-ranked them with the final score generated by BM25 score and semantic similarity score. In TKB48_Run 1 to 3, for each topic, we combined one of the sets of prediction queries and the description as a single question. As shown in Algorithm 1 the top 100 articles were searched cyclically until all prediction problems have been used, and then we removed duplicates and re-ranked by the final score generated by the BM25 score and semantic similarity scores. TKB48_Run 4 just added the title in search queries based on the above process.

3.5.2 Background Linking (Subtopics). We submitted four runs for subtopic tasks. The retrieval conditions are shown as table 2. S_{STC} refers to the semantic similarity score between subtopic and document content.

4 RESULT

We compared our results with the mean and max value with all submitted runs as shown in Figure 1. None of our official runs show significance improvement over the mean value of all teams. Combining with Table 1, in general, temporal recency features applying to all topics did not improve the ranking performance. This indicated us a forehead judgement of temporal topics should be done. For temporal insensitive topics (atemporal topics), temporal recency could harm the performance of useful components like BM25.

The factor of semantic similarity between topic keywords and contents' keywords also contributes to the rank results. This verify that extracted keywords represent the central contents of documents. Comparing the semantic similarities of keywords is a effective way to judge of background articles value.

[4]https://solr.apache.org/

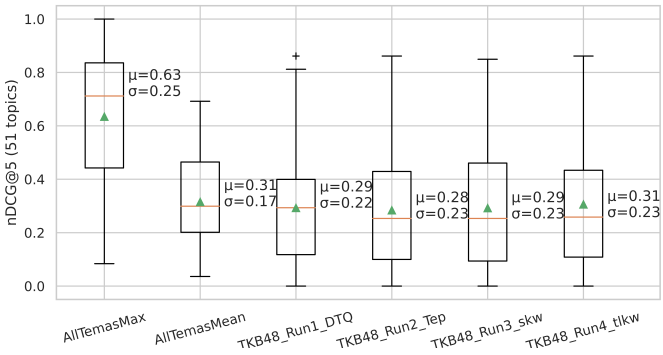

Figure 1: Official runs' performance evaluated in nDCG@5

Table 3: Unofficial Runs Explanation

unofficial runs	retrieve condition		
	query components	*re-ranking factors*	*index fields*
Baseline1	Topics' desc	None	Title, Content
Baseline2	Same as above	None	+ Predicted queries + Key words
Baseline3	Same as above	S_{DC}	Same as above
Baseline4	Same as above	S_{DC}, S_{DK}	Same as above
ltr	+ Predicted Queries	S_{KK}, S_{TK}, S_{TQ}	Same as above
rerank	Same as above	S_{KK}, S_{TK}, S_{TQ}	Same as above

Adding the topic title to the queries is helpful to the performance. Titles in search queries may enhance proportion of useful tokens and optimize the BM25 results.

5 UNOFFICIAL RUNS

We conducted several experiments as illustrated in table 3 after the submission and evaluated using official scripts. To demonstrate the effectiveness of document expansion, we added baseline1 and baseline2. To demonstrate each of the re-ranking factors, we added baseline3 and baseline4. S_{KK}, S_{TK}, S_{TQ} refer to semantic similarity score between keywords of query and doc; keywords and titles; titles and predicted queries. In the ltr run, we trained a temporal topic classifier and a learning to rank model to take better advantage of document freshness and semantic re-ranking factors. During the official runs, we found a forehead classifier should be applied to decide whether a topic is temporal sensitive or not. Topics asking about definitions and comparisons between several objects will not be benefit by temporal freshness. Thus we fine-tuned a temporal topic classifier based on BERT sentence classification task using topics from last year's news track as well as a test collection which is NTCIR Temporalia task[8]. We first compared experiments with and without temporal recency on 2020's topics and assumed that those performances got improved are temporal, and others got decreased are atemporal. Since half of the topics in 2020 do not have a publish date and can not be judged, we used extra data set (Temporalia) to expand training data. Temporalia has annotated temporal and atemporal topics, which is suitable for the training task. The output of the classifier was then used as the feature in a learning to rank model.

The learning to rank model we adopted is one of the listwise models named LambdaMart[17]. We trained it by last year's qrels and using nDCG@5 as a training metric. The imported features are the temporal label of topics which is the output of temporal topic classifier, whether the topic has publish date, semantic similarity scores of keywords, titles and predicted queries, document length, and BM25 score.

Algorithm 2 Re-ranking

```
qtopic ← required query topic
W_R ← temporal recency calculated by equation 1
S_KK ← semantic score between keywords of query&doc
S_TK ← semantic score between title & keywords of doc
S_TQ ← semantic score between title & predicted queries
IsTempo ← TempoTopicClf(qtopic)
if IsTempo==True then
    rankscore = W_R+S_KK+S_TK+Norm(S_TQ)+Norm(S_BM25)
else
    rankscore = S_KK+S_TK+Norm(S_TQ)+Norm(S_BM25)
end if
newrank ← Sort(rankscore)
```

In the re-ranking run, we replaced learning to rank model with the methods explained in Algorithm2.

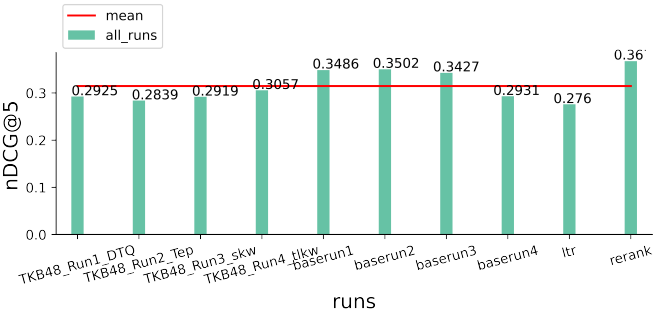

Figure 2: All runs compared with mean score of 51 topics (nDCG@5)

We evaluated extra runs through official scripts. We compared them together with the official runs to demonstrate the effectiveness of document expansion. As shown in Figure 2, from the results, we can observe that indexing the predicted queries and extracted keywords helps retrieve background news articles. No obvious benefit can be observed when we re-ranked results with the semantic similarity between documents' content and topic description or between documents' keyword and topic description. An obvious fall occurs after we added the semantic similarity re-ranking factors to two, which might lower the effect of BM25.

Furthermore, the learning to rank method did not perform better than our designed algorithm. We exported model parameters, the scores of BM25 and semantic similarities of titles occupied around 94.6% in all, and semantic similarity scores of title and keywords as well as title and predicted queries take 5.1%. Thus, temporal features are nearly effective in improving ranking performance.

We analyzed the scores for each topic compared to the mean of all submitted runs. The effectiveness of temporal recency weight is case by case, which indicates us a temporal topic classifier should be added before considering ranking by document freshness. Topics talking about definitions or comparisons between two items are not suitable for this method, while others greatly improved topics concerning upcoming events. However, the experiment of ltr shows temporal features are nearly effective in the ranking process, which may be caused by the lack of training dataset and the sparsity of temporal topics in the task. We compared topics' scores with the max of all submitted runs. Several runs of us have reached or are very close to the max score.

The re-ranking run gives the best results at present. The performance of each topic is relatively better than other runs. The results prove that title, keywords, and predicted queries are more effective than contents or desc in semantic calculations. With a one-tailed paired t-test of the re-ranking run against the Baseline4 and the offcial TKB48_Run4, the difference is significant at $p < 0.05$.

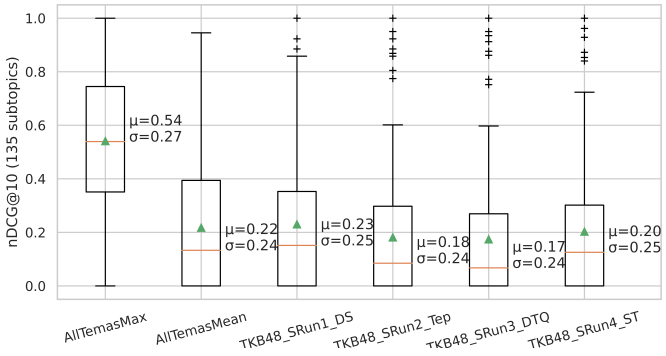

Figure 3: Subtopic runs performance evaluated in nDCG@10

As for the subtopic task, the results further verify that temporal freshness and putting predicted queries into retrieval texts decrease the ranking performance. The performance of each run is shown in Figure 3. No positive significance show between the mean of all teams and submitted runs.

6 CONCLUSION AND FUTURE WORK

This work also adopted a temporal topic classifier and document freshness calculation methods, including documents published before query articles. By analyzing the weight parameters in the learning to rank model, We found temporal features' effectiveness is generally limited. However, document freshness works on several topics that concern upcoming events.

Our future work will focus on fine tuning the docT5query model with last year's dataset. To discover the further ability of the model in the Background linking task.

REFERENCES

[1] Ali Eren Ak, Çaghan Köksal, Kenan Fayoumi, and Reyyan Yeniterzi. 2020. SU-NLP at TREC NEWS 2020. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020 (NIST Special Publication, Vol. 1266)*, Ellen M. Voorhees and Angela

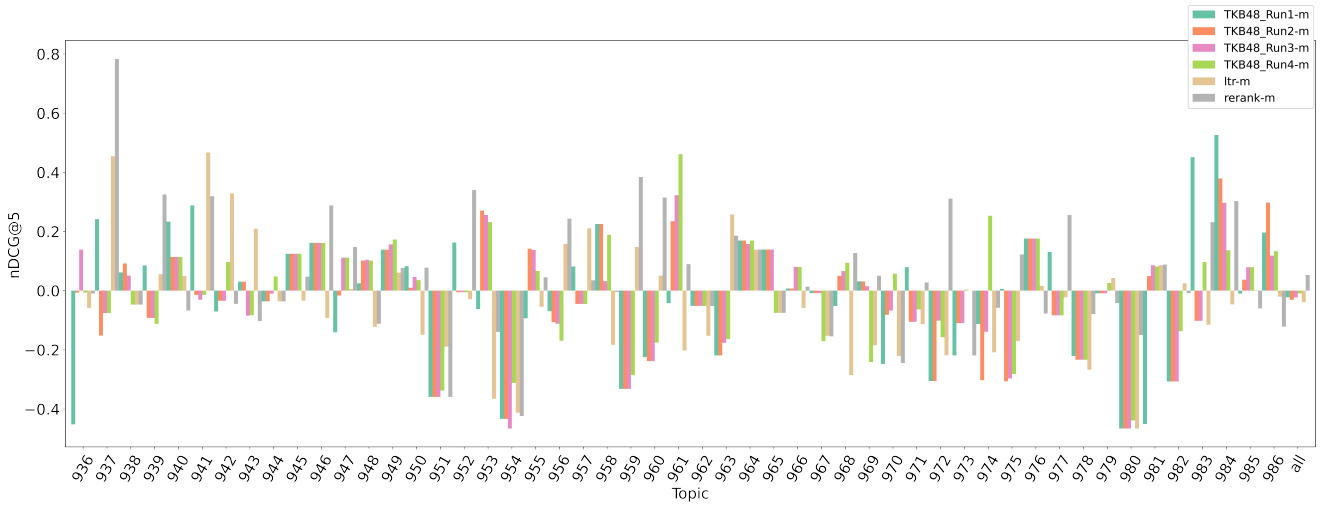

Figure 4: Runs with the mean score of all submitted runs

Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec29/papers/SUNLP.N.pdf

[2] Agra Bimantara, Michelle Blau, Kevin Engelhardt, Johannes Gerwert, Tobias Gottschalk, Philipp Lukosz, Shenna Piri, Nima Saken Shaft, and Klaus Berberich. 2018. htw saar @ TREC 2018 News Track. In *Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018, Gaithersburg, Maryland, USA, November 14-16, 2018 (NIST Special Publication, Vol. 500-331)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec27/papers/htwsaar-N.pdf

[3] Florian Boudin. 2018. Unsupervised Keyphrase Extraction with Multipartite Graphs. arXiv:1803.08721 [cs.IR]

[4] Nathan Day, Dan Worley, and Tim Allison. 2020. OSC at TREC 2020 - News track's Background Linking Task. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020 (NIST Special Publication, Vol. 1266)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec29/papers/OSC.N.pdf

[5] Anup Anand Deshmukh and Udhav Sethi. 2020. IR-BERT: Leveraging BERT for Semantic Search in Background Linking for News Articles. *CoRR* abs/2007.12603 (2020). arXiv:2007.12603 https://arxiv.org/abs/2007.12603

[6] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. TREC Complex Answer Retrieval Overview. In *Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017 (NIST Special Publication, Vol. 500-324)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec26/papers/Overview-CAR.pdf

[7] Ornella Irrera and Gianmaria Silvello. 2021. Background Linking: Joining Entity Linking with Learning to Rank Models. In *Proceedings of the 17th Italian Research Conference on Digital Libraries, Padua, Italy (virtual event due to the Covid-19 pandemic), February 18-19, 2021 (CEUR Workshop Proceedings, Vol. 2816)*, Dennis Dosso, Stefano Ferilli, Paolo Manghi, Antonella Poggi, Giuseppe Serra, and Gianmaria Silvello (Eds.). CEUR-WS.org, 64–77. http://ceur-ws.org/Vol-2816/paper6.pdf

[8] Hideo Joho, Adam Jatowt, Roi Blanco, Hajime Naka, and Shuhei Yamamoto. 2014. Overview of NTCIR-11 Temporal Information Access (Temporalia) Task.. In *NTCIR*.

[9] Pavel Khloponin and Leila Kosseim. 2020. The CLaC System at the TREC 2020 News Track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020 (NIST Special Publication, Vol. 1266)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec29/papers/CLaC.N.pdf

[10] Kuang Lu and Hui Fang. 2018. Paragraph as Lead - Finding Background Documents for News Articles. In *Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018, Gaithersburg, Maryland, USA, November 14-16, 2018 (NIST Special Publication, Vol. 500-331)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec27/papers/udel_fang-N.pdf

[11] Kuang Lu and Hui Fang. 2019. Leveraging Entities in Background Document Retrieval for News Articles. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019 (NIST Special Publication, Vol. 1250)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec28/papers/udel_fang.N.pdf

[12] Sondess Missaoui, Andrew MacFarlane, Stephann Makri, and Marisela Gutierrez-Lopez. 2019. DMINR at TREC News Track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019 (NIST Special Publication, Vol. 1250)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec28/papers/cityuni.News.pdf

[13] Rodrigo Nogueira. 2019. From doc2query to docTTTTTquery.

[14] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. arXiv:1904.08375 [cs.IR]

[15] Ian Soboroff, Shudong Huang, and Donna K. Harman. 2020. TREC 2020 News Track Overview. In *TREC*.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv:1706.03762 [cs.CL]

[17] Qiang Wu, Christopher Burges, Krysta Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Inf. Retr.* 13 (06 2010), 254–270. https://doi.org/10.1007/s10791-009-9112-1