IRCologne at TREC 2021 News Track
Relation-based re-ranking for background linking

BJÖRN ENGELMANN, TH Köln (University of Applied Sciences), Germany
PHILIPP SCHAER, TH Köln (University of Applied Sciences), Germany

This paper presents our approach to the background linking task of the TREC 2021 News Track. The background linking task is to find a set of relevant articles in the Washington Post dataset containing helpful background information for a given news article. Our approach involved a two-stage retrieval process. In the first stage, the 200 most relevant documents were extracted from the entire corpus using BM25. The second stage involved re-ranking using similarity scores based on entities and relations extracted from the query document and the associated 200 relevant documents. For this task, we submitted five runs, each giving different weights to the entities and relations. Our best run received a nDCG@5 of 0.4423, and we were thus able to show that re-ranking with the use of relations leads to a slight improvement over the baseline without re-ranking.

1 INTRODUCTION

The number of online news offerings and their use has been increasing rapidly in the recent past. News has been a big topic in IR for a long time, but the news consumer has not been addressed sufficiently. From the user's point of view, it is still challenging to find one's way through the flood of information. Since the level of knowledge of users varies greatly, it is often essential to get relevant background information to understand and correctly contextualize news articles. However, this background information is difficult to find without assistance. Therefore, the background linking Task exists to develop methods that use approaches from information retrieval to support the user with the massive amount of data.

For a given news article (query document), a ranked set of documents should be returned containing as much relevant background information as possible. Human assessors perform the actual relevance assessments. The relevance score for an article ranges from 0 (The linked document provides little or no useful background information) to 4 (The document must appear in an explanation box or a list of context links, otherwise important context is missing).

In the last iteration of the News Track, several ideas were presented to evaluate the similarity of articles [4]. Document similarity was then used as a measure of relevance. Approaches were presented that used named entities, relation graphs, and embeddings as features of the documents [2, 3, 6, 7]. Surprisingly, it turns out that embeddings seem not well suited for this task. Instead, standard approaches like BM25, achieved the best results [6].

The goal of this submission was to evaluate how re-ranking with named entities and relations affects retrieval performance. We manually examined the relevance scores of News Track 2020 and hypothesized that similar relations across multiple articles are an indicator of relevant documents. A concise example is the relation "goes to" between the entities Nixon and China. Many documents marked as highly relevant contained a variation of this relation. However, many non-relevant articles included the entities China and Nixon but in a different context. Since the meaning of the relation "goes to" can present itself in many variations (e.g., Nixon "visits" China), this work has tried not to exclude other relation variations.

2 DATA

2.1 TREC Washington Post Corpus

The Washington Post dataset (version 4) contains 728,626 documents of news articles or blog posts from 2012 to 2020. The articles are stored in JSON format, and include [1]:

- title
- byline
- date of publication
- kicker (a section header)
- article text broken into paragraphs
- links to embedded images and multimedia (for 2012-2017 documents)

In 2021, the background linking task contained 51 annotated query documents (topics), including a set of subtopics for the subtopic run.

2.2 Preprocessing

As an interface to the indexing pipeline, we used the python package "ir-datasets" to efficiently iterate over the documents, examine individual documents, and avoid encoding problems [8]. To index the articles we used the framework PyTerrier [9], which is a python API for Terrier [10].

All articles that were labeled "Opinion", "Opinions", "Letters to the Editor", or "The Post's View" in the kicker section were filtered out in advance. Furthermore, we filtered out duplicates by identifying identical documents via the URL of the article during retrieval. The fields "title", "author", "kicker", "body" and "body_paras_html" were used as document representation for the query and all candidate articles. Then only the terms of the resulting indexed text were used for comparison during retrieval.

3 METHODS

In this section, we describe the methods we used for the submitted runs. All the experiments we did to evaluate the methods were based on the Washington Post dataset version 3 from 2020.

The flow of the individual modules is shown in Figure 1, whereby only the TF-IDF scores were used for the baseline model. The estimation of the relation scores is described in subsection 3.3.

3.1 Baseline

Since our approach aims to evaluate the effect of incorporating named entities and relations to estimate relevance, a baseline is needed for the retrieval process, pre-selecting documents for the re-ranking step. To evaluate the methods from subsection 3.2 and subsection 3.3, the retrieval performance of the baseline is used as a reference. We used Okapi BM25 [12] as the scoring function for retrieval. Our system used the default parameters and the implementation of PyTerrier. Each query article was transformed into a set of query terms (as described in subsection 2.2) to obtain a similarity score of all documents in the corpus using the scoring function. Only the top 200 retrieved documents were then used for further processing with the re-ranking since the methods for re-ranking are much more computationally intensive, and it is infeasible to re-rank across all the documents from the corpus.

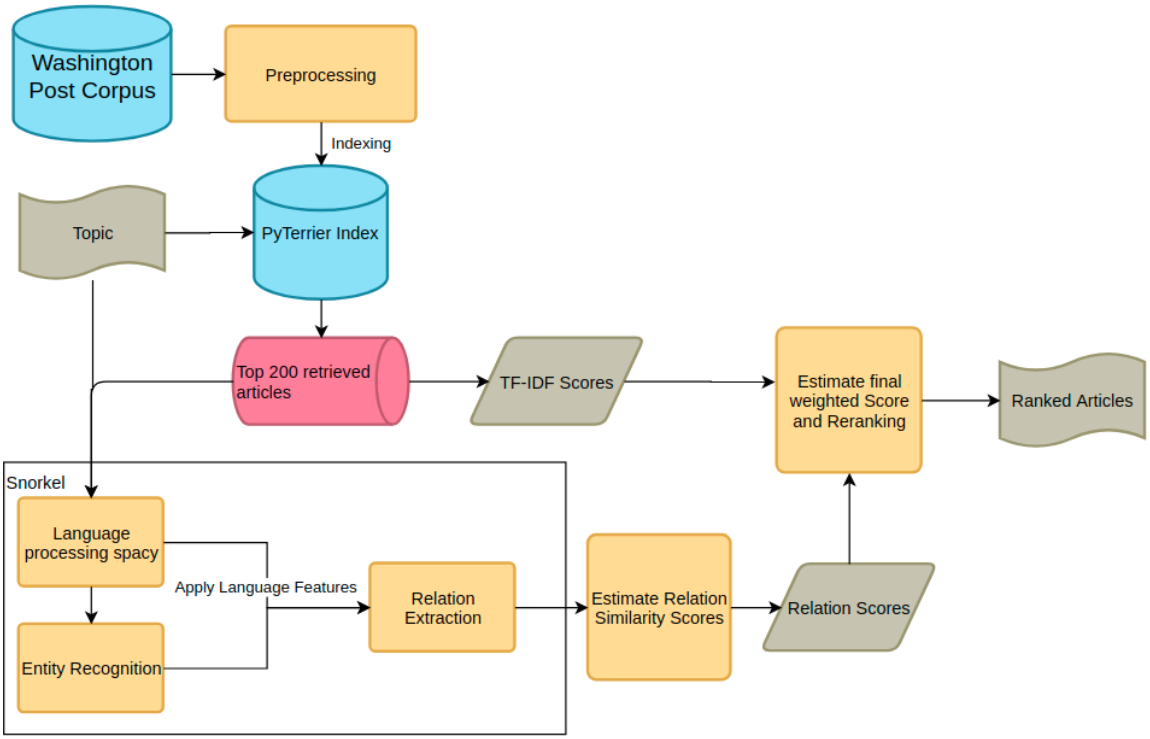


Fig. 1. Retrieval pipeline overview.

3.2 Entity Recognition

For each query, we extracted the entities from the query document q and every document $d \in D$, where D are the top 200 retrieved documents. For named entity recognition, we used the python library spaCy [5]. Only terms corresponding to one of the following entity types were used: 'EVENT', 'FAC', 'GPE', 'LAW', 'LOC', 'NORP', 'ORG', 'PERSON', 'PRODUCT', 'WORK-OF-ART'. These types were selected because they seemed relevant to us through a manual examination of the articles, especially in the news context. We applied the following formula for all pairs of query documents and the 200 corresponding candidate documents to build a similarity score based on the extracted entities:

$$S_{en}(q,d) = \sum_{t \in E_{q\_d}} \log(N/idf(t)) \cdot \log(tf(t)+1). \tag{1}$$

E_{q_d} represents the set of all entities that occur in the query document q and the candidate document d. N is the number of documents in the corpus. The function $idf(t)$ corresponds to inverse document frequency of the term t in the corpus and the function $tf(t)$ to the number of occurrences of t in q. The similarity score was then combined with the score of the baseline model (BM25):

$$S_{ben}(q,d) = bm25(q,d) + \lambda_{en} \cdot S_{en}(q,d). \tag{2}$$

3.3 Relation Extraction

A framework was used to extract relations between entities to apply simple heuristics that link entities when they match a specific predefined pattern. The Snorkel framework [11] applies several so-called labeling functions to pairs of entities to decide whether they are related. These functions serve as a rule of thumb and provide a weak signal for each pair as to whether they are related. The combination of the individual signals then leads to a noisy label. We have defined the following rules for relation identification:

- Both entities appear in the same sentence
- One entity appears as the subject and the other as the object in the sentence
- There is a verb that connects both entities

For this purpose, the previously extracted entities (subsection 3.2) and their grammatical properties were used, with which spaCy annotates the entities. The shared relations of two documents were then used for the similarity score. To integrate also weak relations, we have examined the following cases of relations:

- The entities of the shared relation must be connected by the same verb
- The entities of the shared relation must be connected by any verb
- The entities of the shared relation must appear only in the same sentence

The relation similarity score for documents d and q were then determined (analogous to subsection 3.2) in the following way:

$$S_r(q,d) = \sum_{(t_1,t_2)\in R_{q\_d}} (\log(N/idf(t_1)) + \log(N/idf(t_2))) \cdot \log(k_{t_{12}} + 1). \tag{3}$$

R_{q_d} represents the set of shared entity tuples from q and d. $k_{t_{12}}$ is the number of occurrences of the relation (t_1, t_2) in q. The relation similarity score was then combined with the score of the baseline model (BM25):

$$S_{br}(q,d) = bm25(q,d) + \lambda_r \cdot S_r(q,d). \tag{4}$$

The combination of scoring functions was estimated as follows:

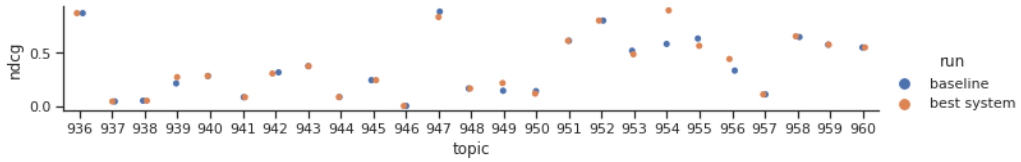
$$S_{benr}(q,d) = bm25(q,d) + \lambda_{en} \cdot S_{en}(q,d) + \lambda_r \cdot S_r(q,d). \tag{5}$$

4 RESULTS

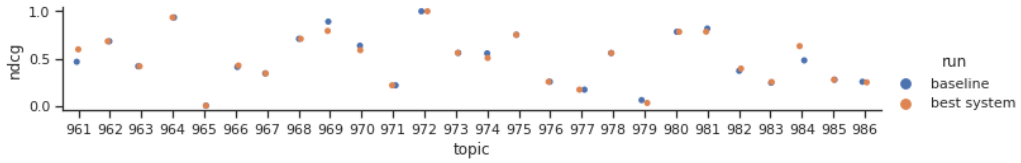
4.1 Background Linking

All experiments were conducted on the 2020 Washington Post dataset (version 3) using the supplied topics and relevance-scored qrels. We performed all the following experiments on the runs with relation extraction only for the entities' scenario in the same sentence. We did not consider the other scenarios in detail because too few common relations were found in the documents. Since we wanted to compare the effect of the entity similarity score, relation similarity score, and a combination with both to the baseline, we evaluated three grid search instances. To avoid overfitting, we used an 80/20 split. The following parameters were evaluated in the respective grid searches:

- Entity run: $\lambda_{en} \in \{0.1, 0.2, \ldots, 1.0\}$
- Relation run: $\lambda_r \in \{0.1, 0.2, \ldots, 1.0\}$
- Combined run: $(\lambda_{en}, \lambda_r) \in \{0.1, 0.2, \ldots, 1.0\} \times \{0.1, 0.2, \ldots, 1.0\}$



(a) topics 936-960



(b) topics 961-986

Fig. 2. Results of all topics for baseline and the best system.

Parameters	Mean nDCG@5 2020	Mean nDCG@5 2021
λ_{en}=0.0, λ_r = 0.0 (base)	0.5404	0.4336
λ_{en}=0.0, λ_r = 0.7	**0.5499**	0.4388
λ_{en}=0.7, λ_r = 0.0	0.5414	0.4377
λ_{en}=0.5, λ_r = 0.2	0.5458	0.44
λ_{en}=0.2, λ_r = 0.5	0.5488	**0.4423**

Table 1. Background linking evaluation results für Washington Post dataset from 2020 (version 3) and 2021 (version 4).

Figure 2 shows a comparison of the baseline run with the best system (λ_{en}=0.2, λ_r = 0.5). It can be seen that the results for most topics do not differ. Only in some cases is there an improvement compared to the baseline (e.g., topic 954 or topic 961).

Table 1 shows the results for the Washington Post dataset from 2020 and 2021. The retrieval model, which considers both entities and relations, provides the best results for the background linking task of 2021. Compared to the previous year, the results are significantly worse. The Mean nDCG@5 score of the baseline decreased by 0.1068 points. Both for the task of 2020 and 2021, it can be seen that the consideration of relations leads to a better retrieval result.

In the following, an example is shown in which the incorporation of the relations resulted in an advantage over the baseline. For topic 984 with the title *Lidl*, an improvement of the nDCG@5 of 0.1513 could be achieved. One relevant document was not ranked among the top 5 by the baseline system. The best system, on the other hand, placed this document with the title *What is Lidl? "5 things the German grocer is bringing to America"* to rank 1.

The five most relevant extracted entity pairs coming from both documents are:

- ('Lidl', 'U.S.')
- ('Brendan Proctor', 'U.S.')
- ('Brendan Proctor', 'Lidl')
- ('German', 'Lidl')
- ('U.S.', 'Arlington')

4.2 Subtopics

Run	Avg. participant runs	λ_{sub}=0.25	λ_{sub}=0.5	λ_{sub}=1	λ_{sub}=2	λ_{sub}=4
Avg. nDCG@10	0.2177	0.2553	0.2698	0.2931	0.3157	0.3343

Table 2. Results for the subtopic task. The first column shows the average nDCG@10 score of all participant runs. The following columns show the results of our approach.

To find relevant documents for the subtopics, we used our baseline approach and added the text of the associated subtopic to the query. The score for the subtopic text was then weighted and included in the final score for the ranking:

$$S_{sub}(q, q_{sub}, d) = bm25(q, d) + \lambda_{sub} \cdot bm25(q_{sub}, d). \tag{6}$$

Here q_{sub} is the text from the subtopic. Since this task was called for the first time this year, there was no training data yet. We, therefore, chose the following weights for the five runs: $\lambda_{sub} \in \{0.25, 0.5, 1, 2, 4\}$.

In Table 2 it can be seen that with increasing λ_{sub} a better retrieval result occurs. Of course, an evaluation with even larger λ_{sub} was not possible without training data, but a weighted subtopic retrieval approach seems reasonable.

5 CONCLUSION

This paper presents an approach that combines classical retrieval approaches, namely BM25, with a similarity ranking based on entities and relations. It turned out that a re-ranking that takes relations into account leads to better results than re-ranking with entities alone. Thus, even if the results do not differ much from the baseline, it is reasonable to assume that relations provide an additional benefit to identify relevant background information. We suspect that this is because the context in which entities are related to each other is better taken into account by relations than by simple matching in the whole document. Future work could consider more complex types of relations or exclude relation types that are not essential for extracting background information.

REFERENCES

[1] Trec washington post corpus. https://trec.nist.gov/data/wapost/. Accessed: 2021-20-15.

[2] A. E. Ak, Ç. Köksal, K. Fayoumi, and R. Yeniterzi. Su-nlp at trec news 2020.

[3] N. Day, D. Worley, and T. Allison. Osc at trec 2020-news track's background linking task. *TREC [30]*.

[4] R. Gautam, M. Mitra, and D. Roy. TREC 2020 NEWS track background linking task. In E. M. Voorhees and A. Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL https://trec.nist.gov/pubs/trec29/papers/IRLABISI.N.pdf.

[5] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020. doi: 10.5281/zenodo.1212303.

[6] P. Khloponin and L. Kosseim. The clac system at the trec 2020 news track. In *TREC*, 2020.

[7] K. Lu and H. Fang. Aspect based background document retrieval for news.

[8] S. MacAvaney, A. Yates, S. Feldman, D. Downey, A. Cohan, and N. Goharian. Simplified data wrangling with ir-datasets. In *SIGIR*, 2021.

[9] C. Macdonald and N. Tonellotto. Declarative experimentation ininformation retrieval using pyterrier. In *Proceedings of ICTIR 2020*, 2020.

[10] C. Macdonald, R. McCreadie, R. L. Santos, and I. Ounis. From puppy to maturity: Experiences in developing terrier. *Proc. of OSIR at SIGIR*, pages 60–63, 2012.

[11] A. Ratner, S. H. Bach, H. R. Ehrenberg, J. A. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. *CoRR*, abs/1711.10160, 2017. URL http://arxiv.org/abs/1711.10160.

[12] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94*, pages 232–241. Springer, 1994.