Semantic Search for Background Linking in News Articles

Udhav Sethi
University of Waterloo
udhav.sethi@uwaterloo.ca

Anup Anand Deshmukh
University of Waterloo
aa2deshmukh@uwaterloo.ca

ABSTRACT

The task of background linking aims at recommending news articles to a reader that are most relevant for providing context and background for the query article. For this task, we propose a two-stage approach, IR-BERT, which combines the retrieval power of BM25 with the contextual understanding gained through a BERT-based model. We further propose the use of a diversity measure to evaluate the effectiveness of background linking approaches in retrieving a diverse set of documents. We provide a comparison of IR-BERT with other participating approaches at TREC 2021. We have open sourced our implementation on Github[1].

Author Keywords

Natural Language Processing; Information Retrieval; BERT; Background Linking; TREC

INTRODUCTION

Online news services have become key sources of information and have affected the way we consume and share news. While drafting a news article, it is often assumed that the reader has sufficient information about the article's background story. This may not always be the case, which warrants the need to provide the reader with links to useful articles that can set the context for the article in focus. These articles may or may not be by the same author, can be dated before or after the query article, and serve to provide additional information about the article's topic or introduce the reader to its key ideas. However, determining what can be categorized as an article providing background context and retrieving such documents is not straightforward.

Motivated by this problem, the background linking task was introduced in the news track of TREC 2018. This task aims to retrieve a list of articles that can be incorporated into an "explainer" box alongside the current article to help the reader understand or learn more about the story or main issues contained therein.

In this paper, we propose a two-stage approach, IR-BERT, to address the problem of background linking. The first stage filters the corpus to identify a set of candidate documents which are relevant to the article in focus. This is achieved by combining weighted keywords extracted from the query document into an effective search query and using BM25 [9] to search the corpus. The second stage leverages Sentence-BERT [8] to learn contextual representations of the query in order to perform semantic search over the shortlisted candidates. We hypothesize that employing a language model can be beneficial to understanding the context of the query article and helping identify articles that provide useful background information.

This paper is structured as follows: In section 2, we provide an overview of prior work that motivates our strategies. In section 3, we describe in detail our retrieval approach, followed by sections 4 and 5, where we describe our experiments and discuss the retrieval performance of our method. Finally, we summarize and conclude our work in section 6.

RELATED WORK

BM25 [9] is one of the most popular ranking functions used by search engines to estimate the relevance of documents to a given search query. It is based on a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of their proximity within the document. Several previous approaches to background linking are built using BM25. The Anserini toolkit developed by Yang et al. [13] further standardizes the open-source Lucene library. Along with BM25, it has been used to effectively tackle the background linking problem [12]. Another set of approaches, ICTNET [4] and DMINR [5], leverage the use of named entities in the query article to build a search query for BM25.

Previous work has also exploited language models such as BERT for the task of ad-hoc retrieval [13, 7]. BERT [2] is pre-trained on large open-domain textual data and has achieved state-of-the-art results in many downstream NLP tasks. It has also proven to be an effective re-ranker in many information retrieval tasks. For example, Dai and Callan [1] showed that employing BERT leads to significant improvements in retrieval tasks where queries are written in natural languages. This is a direct consequence of their ability to better leverage language structures. Nogueira and Cho [6] used BERT on top of Anserini to re-rank passages in the TREC Complex Answer Retrieval (CAR) task [3]. Similar re-ranking mechanisms have also shown promise in open-domain question answering [14].

The task of semantic search is very relevant to ad-hoc retrieval where queries are written in natural languages. There are two main issues with using BERT for finding the semantic sim-

[1] https://github.com/Anup-Deshmukh/TREC_background_linking

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

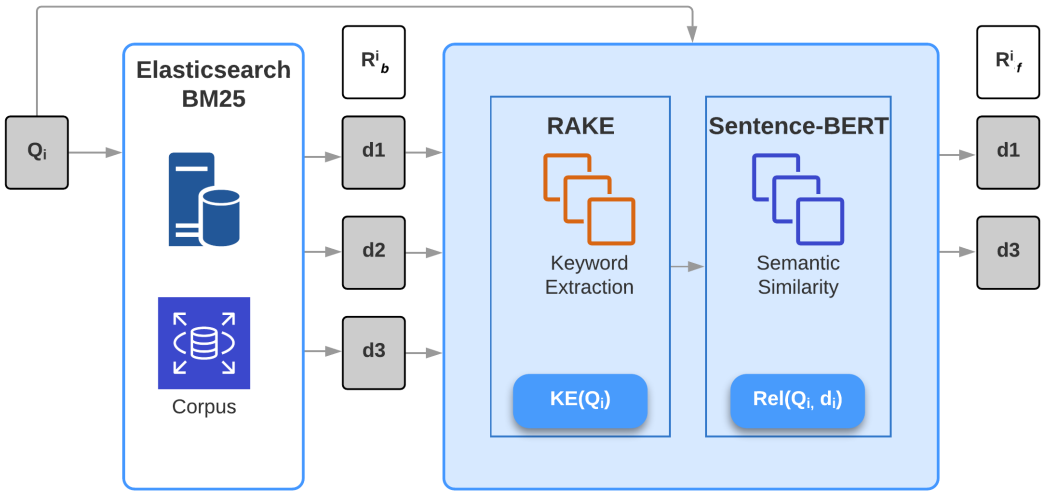

Figure 1. Stages of retrieval in the IR-BERT pipeline

ilarity between text documents. First, to compare a pair of documents, both need to be input to the model, which causes significant computational overhead at inference time. Second, to solve the problem of semantic search, a widely used approach is to map documents into the vector space where similar documents are closer. Common practices such as averaging the BERT output layer or using the output of [CLS] token yield poor embeddings. Sentence-BERT [8] alleviates these issues by leveraging siamese and triplet network structures. It derives semantically meaningful sentence embeddings that can be compared using cosine-similarity. In this work, we leverage Sentence-BERT as a part of our architecture.

METHODOLOGY

The background linking task can be formulated as following: *Given a news story S and a collection of news articles A, retrieve other news articles from A that provide important context or background information about S.*

It is reasonable to consider this task as a specific case of news recommendation aimed at retrieving relevant articles from a corpus (A) for a query generated from an article (S). We hypothesize that most of the articles that can provide contextual information about the query article were likely published before it. To this end, we filter out forward links from our results, i.e., the articles published after the query article are not considered.

IR-BERT attempts to solve the problem of background linking in two stages. In the first stage, we construct a weighted query Q_i from article S_i and use BM25 to retrieve a set of p candidate documents. Let this set of documents be $R_b^i = \{d_1, d_2, .., d_p\}$ where $|R_b^i| = p$. In the second stage, we conduct a semantic search of Q_i over the set of retrieved documents R_b^i to arrive at the final set of documents $R_f^i = \{d_1, d_2, .., d_t\}$ where $|R_f^i| = t$. The two stages of IR-BERT are illustrated in Figure 1.

Weighted Search Query + BM25

We first build an effective search query that best captures the relevant topics of the query article. The problem is formulated as extracting the essential keywords from the query article, assigning them weights according to their relevance, and concatenating them to form a query. This query is then used to search the corpus using BM25, through which a ranked list is generated.

To find the keywords $\{k_1, ..., k_n\}$ from a query document S, we sort all the words in S in decreasing order of their TF-IDF score. To assign different relevance scores to the keywords, we define a weight w_j for each keyword k_j as follows:

$$w_j = nint\left(\frac{s_j}{\sum_{k=1}^{n} s_k} \cdot n\right) \tag{1}$$

$$s_j = \text{TF}(k_j, S) \cdot \text{IDF}(k_j, A) \tag{2}$$

where n is the number of keywords, and TF and IDF are the two statistics, term frequency and inverse document frequency, respectively. To apply the weight for each keyword, we round its value to the nearest integer w_j and repeat the jth keyword k_j, w_j number of times in the query. We also assign different weights to the contribution of keywords in the title and body of the article. This weighted query is fed to BM25, and the top p retrieved articles (R_b^i) are selected as candidate documents.

Semantic Search using BERT

The first stage uses BM25 to retrieve candidate documents from the corpus entirely based on the term frequencies of words appearing in the query article. To understand the context of the query article, it is important to take the semantics of words into consideration. This is because the background articles may not necessarily contain the same keywords as the search query constructed from the query article. For example, a query article *In Russia, political engagement is blossoming online* is likely to have **Russia** and **online** in the constructed query. In order to find the background articles, the retrieval

model first must understand that **Russia** is a *country* and **online** refers to *social media* platforms like *Facebook* and *Twitter* which are based on the *internet*. To this end, we use a BERT-based model in the second stage to gain semantic knowledge of the query and candidate articles.

RAKE: Before carrying out the semantic search over the set of documents R_b^i, it is important to feed only those words to sentence-BERT whose semantic meaning could benefit us. Thus, every document in R_b^i is passed through the Rapid automatic Keyword Extraction (RAKE) algorithm [10]. RAKE takes a list of stopwords and the query as inputs and extracts keywords from the query in a single pass. RAKE is completely domain independent and hence can be utilized for our specific newswire domain. It is based on the idea that co-occurrences of words are meaningful in determining whether they are keywords or not. The relations between the words are hence measured in a manner that automatically adapts to the style and content of the text. This allows RAKE to have adaptive measurement of word co-occurrences which are used to score candidate keywords.

Sentence-BERT (SBERT): Sentence-BERT [8] is a modification of the pretrained BERT network that adds a pooling operation on top of the last layer of BERT and is fine tuned to derive a fixed size sentence embedding. Sentence-BERT further uses siamese and triplet network structures [11] to update the weights of this model. The siamese network allows for sentence embeddings that are specifically trained to work with a similarity measure like cosine-similarity. Figure 2 illustrates the Sentence-BERT architecture at inference time.

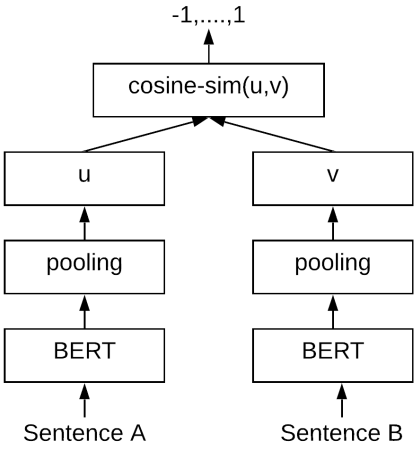

Figure 2. Sentence-BERT architecture at inference to calculate similarity scores

SBERT is used to obtain the embeddings for the query document Q_i and each of the documents in R_b^i using their keywords extracted by RAKE. The documents in R_b^i are then sorted according to their cosine similarity with the query Q_i using equation 3 (where e_1 and e_2 are the embeddings of the two documents being compared). Algorithm 1 outlines the steps involved in generating the final list of documents R_f^i via SBERT embeddings.

$$\text{CosineSim}(e_1, e_2) = \frac{e_1.e_2}{||e_1||.||e_2||} \tag{3}$$

Algorithm 1 RerankCandidates(Q_i, R_b^i)

$p \leftarrow$ Number of documents retrieved by BM25
$t \leftarrow$ Required number of final documents
$q_i \leftarrow$ SBERT(Q_i)
for j = 1, . . . p **do**
  E_j = SBERT($R_{b,j}^i$)
  f_j = CosineSim(E_j, q_i)
end for
$F \leftarrow R_b^i$ sorted by decerasing f_j
$R_f^i \leftarrow$ top t documents in F
return R_f^i

EXPERIMENTS

Dataset

We used the Washington Post Corpus[2] released by TREC for the 2021 news track to compare our approach with runs by other participants in TREC 2021. The collection contains 728,626 news articles and blog posts from January 2012 through December 2020. For other experiments, we used the Washington Post 2018 corpus, which is a subset of the 2021 dataset, containing 608,180 articles from Januray 2012 up to August 2017. Both datasets were preprocessed using the steps shown in Figure 3. The articles are in JSON format, and include fields for title, date of publication, kicker (a section header), article text, and links to embedded images and multimedia.

Our method relies on Elasticsearch[3] as the indexing platform. During indexing, we extracted the information from the various fields and indexed them as separate Elasticsearch fields. We also created a new field to store the body of the article. For this, we first extracted the HTML text content from the fields marked by type 'sanitized_html' and subtype 'paragraph', and then concatenated them after using regular expressions to extract the raw text from HTML text. Next, we performed lower-casing, stop-word removal, and stemming on the raw text. The final preprocessed text was then indexed as a separate text field in Elasticsearch, representing the article body. While indexing, we removed the articles from the "Opinion", "Letters to the Editor", and "The Post's View" sections (as labeled in the "kicker" field) as they are as they are deemed irrelevant according to the TREC guidelines.

Setup

We used the the default scoring method in Elasticsearch, BM25, as the retrieval model. To set the relative weights for the title and body of the article in the search query, we leveraged Elasticsearch boosting queries. We experimented with different combinations of a number of parameters, the final values for which are listed in Table 1.

[2] https://trec.nist.gov/data/wapost/
[3] https://www.elastic.co/

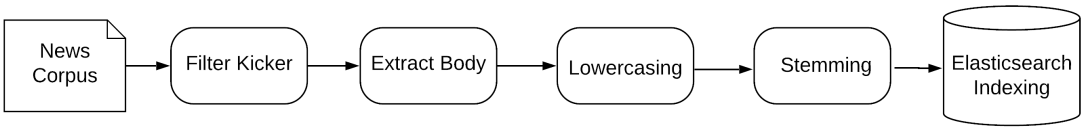

Figure 3. Data Preprocessing Steps

# words in constructed query Q	100
# filtered results from BM25	180
# keywords generated from RAKE	100
% keywords in Q from title	70
% keywords in Q from body	30

Table 1. Parameter Values

Methods	nDCG@5
KWVec	**0.4620**
IR-Cologne	0.4423
TKB48-DTQ	0.2925
FUH-N	0.2655
IR-BERT	0.3613

Table 2. nDCG@5 scores of IR-BERT and other participating methods on Washington Post 2021 dataset

Evaluation Metric

The primary metric used by TREC for the background linking task is nDCG@5 with a gain value of 2^{r-1}, where r is the relevance level, ranging from 0 (provides little or no useful background information) to 4 (provides critical context). The zero relevance level contributes no gain.

Diversity Measure

As per TREC guidelines, one of the criteria for ranking for the background linking task is to have a retrieved list of articles which are diverse. The idea of diversity may seem subjective but it is possible to formulate it as given in equation 4.

$$Diversity = \frac{1}{|Q|}\sum_{Q_i}\frac{1}{|R^i_f|}\sum_{a\in R^i_f}\sum_{b\in R^i_f, b\neq a} dist(d_a, d_b) \quad (4)$$

For every retrieved document list R^i_f, we calculate the sum of distances between all possible pairs of documents d_a and d_b. The distances between representation of documents can be captured by metrics like cosine similarity. These distances are then summed over all queries/topics Q to get the diversity score.

RESULTS

We review the relative performance of IR-BERT against some of the other participating methods at TREC 2021 in Table 2. Results show that IR-BERT performs better than TKB48 which utilizes transformer based Doc2Query re-ranker. IR-BERT also beats methods like FUH-N and QU which carry out matrix based indexing operations and transfer learning from subtopics respectively[4]. On the other hand, methods like KWVec and IR-Cologne achieve higher nDCG@5 scores than IR-BERT. KWVec is similar to IR-BERT in using Sentence-BERT and Elasticsearch. IR-Cologne uses extracted entities and relations for reranking.

To investigate the effects of employing a language model for the background linking task, we compare the performance of alternate architectures on the Washington Post 2018 corpus. We list the nDCG@5 and nDCG@10 scores for each of these approaches in Table 3. The first two approaches utilize only stage 1 of our architecture, i.e., they simply build a search query and use BM25 for retrieval. While wBT+BM25 uses only weighted body and title while constructing a query, wQ+BM25 uses also uses weights for all the words present in the query document. We observe that wQ+BM25 gives the best nDCG@10 score, which suggests that articles that provide useful background information are likely to contain keywords similar to the query article. Furthermore, IR-BERT achieves the highest nDCG@5 score, suggesting that contextual understanding of the article's story can benefit the background linking task. It is also interesting to note that using RoBERTa on top of BM25 for re-ranking harms the performance compared to using the vanilla BERT model.

Methods	nDCG@5	nDCG@10
wBT+BM25	0.4088	0.4155
wQ+BM25	0.3942	**0.4315**
IR-RoBERTa	0.394	0.3918
IR-BERT	**0.4199**	0.4104

Table 3. Comparison of nDCG scored for alternate methods on Washington Post 2018 dataset

In our final set of experiments we compare the diversity of the documents retrieved by all our approaches on the Washington Post 2018 dataset using equation 4 (See Table 4). We observe that IR-RoBERTa, which performs relatively worse on nDCG measures, retrieves the most diverse list of background articles for a given query.

Methods	Diversity Score
wBT+BM25	0.9067
wQ+BM25	0.912
IR-RoBERTa	**0.921**
IR-BERT	0.9084

Table 4. Comparison of diversity of retrieved documents from various methods on Washington Post 2018 dataset

[4]Although we did not have access to the nDCG@5 score of QU, TREC 2021 news track overview shows the better performance of IR-BERT.

CONCLUSION

In this paper, we described a two-stage approach to solve the background linking task of the TREC 2021 news track. The first stage attempts to extract representative keywords from the query article and uses them to retrieve a candidate set of the background links. The second stage leverages the contextual understanding gained from BERT to perform semantic search over the retrieved candidates. Our model, IR-BERT, achieved an nDCG@5 score of 0.3613 on the TREC Washington Post 2021 dataset. Overall, the participating models and their performance show the effectiveness of re-ranking by leveraging the contextual understanding of transformer based models.

ACKNOWLEDGMENTS

We would like to thank Prof. Gordon V. Cormack, Professor at the School of Computer Science, University of Waterloo, for his guidance during the course of this work.

REFERENCES

[1] Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 985–988.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[3] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. TREC Complex Answer Retrieval Overview. In *TREC*.

[4] Yuyang Ding, Xiaoying Lian, Houquan Zhou, Zhaoge Liu, Hanxing Ding, and Zhongni Hou. 2019. ICTNET at TREC 2019 News Track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC*.

[5] Sondess Missaoui, Andrew MacFarlane, Stephann Makri, and Marisela Gutierrez-Lopez. 2019. DMINR at TREC News Track. In *TREC*.

[6] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[7] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424* (2019).

[8] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[9] Stephen Robertson, Hugo Zaragoza, and others. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[10] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory* 1 (2010), 1–20.

[11] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[12] Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using Lucene. *Journal of Data and Information Quality (JDIQ)* 10, 4 (2018), 1–20.

[13] Peilin Yang and Jimmy Lin. 2018. Anserini at TREC 2018: CENTRE, Common Core, and News Tracks. In *Proceedings of the 27th Text REtrieval Conference (TREC 2018)*.

[14] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718* (2019).