SU-NLP at TREC NEWS 2020

Ali Eren Ak, Çağhan Köksal, Kenan Fayoumi, and Reyyan Yeniterzi^{*}

Sabancı University, İstanbul, Turkey

Abstract

This paper presents our work and submissions for the TREC 2020 News Track: background linking and wikification tasks. For the background linking task, we investigated utilization of transformer-based architectures as either to map news articles to sentence embeddings, or to extract summaries of queried news articles or to re-rank the candidate documents. Even though none of these approaches outperformed our regular full-text baseline search approach, they provided some insights and future research directions. In the wikification task, after extracting and linking the entities to their corresponding entities, we analyzed two ranking approaches; one depends on entities positions within the text and the other depends on vector similarities between the news article and entity's linked page.

1 Introduction

Use of the online news services has been dramatically increasing in the last couple of years. Pew Research studies states that in 2018 93% of American adults get at least some of their news online [1]. Although the common assumption is that the readers have the necessary background knowledge about the context and entities mentioned in any news article, this expectation may not be valid for many readers. TREC News Track is specifically focusing on this problem by introducing two tasks, background linking and wikification. The purpose of the background linking task is to retrieve relevant news articles which help readers to understand the context and obtain background knowledge of a given news article (the query article). On the other hand, the wikification task goes one step further and uses the Wikipedia articles too,

^{*}Corresponding author: reyyan.yeniterzi@sabanciuniv.edu

in order to provide more context to the readers on important short passages within the news article.

We participated to both of these tasks. For the background linking task, we specifically investigated the capabilities of recent transformer-based architectures on representing news articles. We explored different ways to represent the news articles, such as encoding them to vectors or using the extracted summaries. Even though, these proposed approaches did not outperform some classical baseline approaches, they provided some useful insight on their capabilities in this background linking task.

We approached to the wikification task as it is an entity linking and ranking task. Therefore, we started by applying the standard entity linking pipeline. Initially identified the entity mentions by using a NER tool and then retrieved and ranked the entity corresponding Wikipedia pages at the same time by using a search engine. Finally, we ranked the linked Wikipedia pages based on their relevance to the news article. In this final ranking part, we explored both the position of entities in the news article and document vector similarity between the news article and entities' wiki-pages.

2 Dataset

This year, third version of the TREC Washington Post Collection was released. This version is different from its priors as it contains more documents (around 150K) and the near duplicates were eliminated. The same Wikipedia dump from previous years TREC News was also used this year.

The background linking task contains 50 new topics along with their Washington Post articles. These 50 topics are also used for the Wikification task. Topics from previous years were used to fine-tune and evaluate our models during development.

3 Background Linking Task

This is the third year of the background linking task. Last year only one participant group [2] explored BERT for this task. This year we used transformer architectures in different ways to better understand their capabilities in this task. More specifically, we examined the effectiveness of sentence embeddings by creating dense vector representations of passages using the Universal Sentence Encoder. Additionally, we explored the prominent NLP model BERT in two ways, (1) for creating summaries of the topic news articles and (2) for re-ranking news articles.

3.1 Our Approach

In this section, we briefly describe our baseline bag-of-words and transformerbased approaches.

3.1.1 Baseline (FullText)

As a baseline approach, we used the bag-of-words approach with full-text indexing and querying. ElasticSearch's ¹ default settings were used for indexing and ranking. Only a date filter was applied in order to limit the retrieval to the news documents which were published before the queried news article. The same date filtering was used in all the following approaches.

3.1.2 Universal Sentence Encoder (USE)

Universal Sentence Encoder [3] is a neural network model which can encode any varying length text into a 512-dimensional dense vector. Transformer or Deep Averaging Network (DAN) models are used as the encoder architecture.

In this paper, all Washington Post news articles were mapped to their USE vectors by using the available pretrained models ² and then these vectors were indexed with ElasticSearch. Article vectors' cosine similarity scores with respect to the query news article's vector were used to retrieve and rank them.

3.1.3 BERT Summarization (BERT Summ)

The goal of extractive summarization is to extract a subset of sentences from a provided text which are the most important and key sentences for summarizing the corresponding input text. Since background linking task's queries consist of long news articles, extractive summarization can be useful in identification of the useful sub-parts of the text that can be used for querying.

In order to investigate BERT's capabilities on identification of the key sentences from a text, BERT Extractive Summarization $tool^3$ was used. This tool is based on a model which creates sentence embeddings and then clusters them in order to choose the central ones [4]. With this tool, on average 800 word long news articles were reduced to 180 words long summaries. Then

¹https://github.com/elastic/elasticsearch

²https://tfhub.dev/google/universal-sentence-encoder/2

³https://pypi.org/project/bert-extractive-summarizer/

these extracted summaries were used as queries to retrieve and rank the news articles which were indexed as full-text (without the summarization).

3.1.4 BERT Re-Ranking (BERT RR)

Both Universal Sentence Encoder and BERT Extractive Summarization tools were used as they are without any fine-tuning for our background linking task. In order to investigate the effects of task specific fine-tuning, we adapted the next sentence prediction task used during pre-training of BERT. This idea was successfully applied to ad-hoc document ranking before [5, 6]. The authors used to re-rank a set of retrieved candidate documents by using the query-document similarity scores calculated after processing the query as first text segment and document as the next one.

In this paper, we follow the work of CEDR [5]. We used the news articles from our baseline full-text approach as the initial candidates. Later on BERT with a next sentence prediction layer was used to re-rank these news articles based on their similarity to the query news article. In order to better analyze the affects of fine-tuning, this re-ranking part was performed both with finetuning and without it. During the one with fine-tuning, the official 2018 and 2019 background linking relevance judgements (qrels) were used.

As a technical note, since our queries are much longer than regular keyword queries, in order to work around BERT's maximum sequence length, both the query and candidate news articles were cropped or padded to the same length (half of 512). For our experiments, we used HuggingFace BERT's implementation 4 .

3.2 Experiments

All these approaches were applied to both previous years' and this year's background linking topics. The results of these experiments are summarized in Table 1. NDCG@5 is used as a primary metric in background linking task. Our baseline FullText approach performed very similar to the median score of all runs. That is expected since using full text for both indexing and querying is a general baseline.

Among the proposed architectures, the Universal Sentence Encoder returned the lowest scores. There may be several reasons for this. One reason can be that these models encode the semantic characteristics of texts in general but maybe those characteristics are not the right ones for this particular task. For example, many countries perform some type of elections which

⁴huggingface.co/bert-base-uncased

	2018	2019	2020
SUNLP_FullText	0.4049	0.5623	0.5343
SUNLP_USE	0.3308	0.4145	0.3820
SUNLP_BERT_Summ	0.3351	0.5288	0.4447
SUNLP_BERT_RR	0.3480	0.3670	0.4559

Table 1: NDCG@5 Scores of Background Linking Approaches

are discussed in news articles. All these news articles discussing different elections probably contain similar words and context related to the elections in general, such as polls, vote, count etc. Therefore, these news articles are most likely mapped to similar vectors but it is not the case that these elections of different countries are related to each other. Fine-tuning these models for this particular task can return much better results. In our submission, we used the USE model trained with the DAN architecture. Due to the time constraints, experiments with the transformer architecture could not be done. DAN architecture generally returns lower scores compared to the transformers. This may be yet another reason why this system received a lower score compared to other transformer-based approaches.

Extracting useful key sentences with BERT and using those summaries to search relevant news articles generally performed better than USE, but still returned lower scores than the baseline. Even though the summaries got better rankings on 9 topics compared to using the full article as query, it is still far away from a general out-performance. Similar to the USE, the model used in here was not fine-tuned for the background linking task, therefore the selected sentences may be useful overall but may not be the right ones to use for searching the relevant news articles.

BERT re-ranking was applied in two ways, with or without fine-tuning. Each year 50 new topics are released. For 2020 we used both 2018 and 2019 topics (around 100 topics), but in 2018 and 2019 only 50 topics were used. Even with this limited small data, fine-tuned models outperformed the models without fine-tuning. Therefore, we used the fine-tuned models in our submission. Similar to others, BERT_RR could not outperform the baseline but returned competitive results compared other models. An interesting observation is that all other models returned lower scores in 2020 compared to 2019. Only BERT_RR returned a much higher score in 2020 compared to 2019. This may be due to the almost twice the size of the training data used in fine-tuning for 2020 topics. Fine-tuning with more data may provide higher improvements.

4 Wikification Task

This is the first time the wikification task has been introduced as part of TREC News. Previous years' TREC News tracks included entity ranking task in which case the entity lists were given together with the linked Wikipedia pages, and participants were asked to only rank them based on their relevance to the news article. The wikification task is a more end-toend version of this as it includes finding the useful short passages within the news article and their corresponding Wikipedia pages as well. This year we focused on entities only. Normally the task description states that passages can be linked to either Wikipedia pages or other news articles. But in this study we only used Wikipedia pages for entity linking.

4.1 Our Approach

Our approach for the Wikification task consists of two steps. The first step is the identification of possible entity candidates from the news article and using these mentions to retrieve relevant Wikipedia pages for entity linking. The second step is the ranking of these retrieved Wikipedia pages based on their relevance and importance to the news article. For the second step two different approaches were applied and compared.

4.1.1 Entity Extraction and Linking

In this first step, initially the entities were extracted from the news articles. Stanford CoreNLP's NER [7] tool was used to identify the entities. The tagged entities generated by this tool were processed further in order to create better entity lists. Numerical and temporal expressions were ignored. Similarly pronouns and any tagged entities which starts with a lowercase letter were removed from the entity list. Finally, in order to filter the near-duplicate entities such as "Donald Trump", "President Trump", "Trump" or "Candidate Trump"; the frequency of words, their positions and phrase match were used.

After the entity mention lists were constructed, these entities were searched over an index built over Wikipedia articles. ElasticSerch was used to built this Wikipedia index which consists of different Wikipedia fields such as title, content, categories etc. During the search, these different fields were explored in terms of their effects on entity page retrieval's quality. As expected using only the title field of the Wikipedia page outperformed other field combinations in terms of returning better Wikipedia page matches for entities. During these experiments only the top returned Wikipedia page was considered for each entity.

4.1.2 Entity Ranking

As the next step, similar to last year's task, entity linked wiki-pages should be ranked based on their relevance to the news article. Two approaches were investigated in this step. The first approach is based on the position of entities within the news article. Ranking entities based on the order they are mentioned in the article seems to return good results based on previous years' TREC entity ranking task results. We also applied the same idea in our proposed system SUNLP_textorder.

As our second approach, we used the approach from last year's topranking system at the entity ranking task. In that approach, entities were ranked according to the cosine similarity between their linked wiki-pages' vector and news articles' vector. Doc2Vec model [8] was used to create the vector representations for both Wiki-pages and news articles. This approach is referred as SUNLP_doc2vec.

4.2 Experiments

Since this is the first time this task is organized, we did not have any data from previous years to analyze our proposed approaches during the development phase. For the TREC 2020 topics, we received the NDCG@5 scores for both of our runs together with some min/median/max scores of all runs. Our first system, SUNLP_textorder scored 0.3168 on average while last year's top-performing approach, SUNLP_doc2vec, scored 0.2378.

Performances of systems with respect to individual topics are displayed at Figure 1 together with the max and median scores. Overall using the text order for ranking achieved maximum score in 21 out of 45 topics (excluding the topics with maximum score 0). Similarly in 35 of the topics, this approach got higher scores than the median.

Last year's top-performing system performed worse than the text ordering. This may be due to the errors propagated from prior steps and text ordering being a more robust system here. Since this year's task is not exactly the same as last year's, a more detailed analysis is required and will be performed as future work.



Figure 1: Topic-based NDCG@5 Scores of *SUNLP_textorder*, *SUNLP_doc2vec*, and reported max and median

5 Conclusion

In this paper, we presented our work and experiments for the background linking and wikification tasks. In background linking task we mainly explored the effects of different and recent neural network architectures. For the wikification task, we mainly approached to the problem as an entity linking and ranking task. The experiments provided some useful insights but there are still some research questions left for future work.

References

- I. Soboroff, S. Huang, and D. Harman, "Trec 2019 news track overview.," in *TREC*, 2019.
- [2] Y. Ding, X. Lian, H. Zhou, Z. Liu, H. Ding, and Z. Hou, "Ictnet at trec 2019 news track.," in *TREC*, 2019.
- [3] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al., "Universal sentence encoder," arXiv preprint arXiv:1803.11175, 2018.
- [4] D. Miller, "Leveraging bert for extractive text summarization on lectures," arXiv preprint arXiv:1906.04165, 2019.
- [5] S. MacAvaney, A. Yates, A. Cohan, and N. Goharian, "Cedr," Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Jul 2019.
- [6] Z. Dai and J. Callan, "Deeper text understanding for ir with contextual neural language modeling," Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Jul 2019.
- [7] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60, 2014.
- [8] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, pp. 1188–1196, 2014.