Applying Pre-trained Systems to Sentence Retrieval

Yastil Rughbeer[1,2] [0000-0003-1974-2410], Anban Pillay[1,2] [0000-0001-7160-6972] and Edgar Jembere[1,2] [0000-0003-1776-1925]

[1] University of KwaZulu-Natal, Westville 4001, South Africa
yastil350.rughbeer@gmail.com, {pillayw4,jemberee}@ukzn.ac.za
[2] Centre for AI Research (CAIR), Cape Town, South Africa

Abstract. Our team (Yastil_R) made two submissions to the Epidemic Question Answering track of TAC 2020. However, due to an overlooked error in the second submission, this paper only discusses Run_1. Nevertheless, our first submission tackles an important problem in the field of Information Retrieval. Specifically, we highlight that large-scale sentence retrieval is inherently difficult for pre-trained systems. As a result, this work aims to improve the performance of these systems in such environments. To achieve this, we turn our attention to the cascade framework, which represents an established paradigm of applying pre-trained systems to large collections. Notably, the cascade framework consists of two stages. The first stage utilizes non-neural systems to retrieve the top *k* sentences for a given query. A pre-trained system then ranks these sentences from most to least relevant. Due to the sequenced nature of retrieval and ranking, they can be considered as mutually exclusive events. Consequently, we argue that there is a strong correlation between the non-neural and pre-trained system's performance. Hence, although counterintuitive, our work focused on improving the non-neural system's performance. We achieved this by assuming that retrieving passages instead of sentences would increase the likelihood of retrieving sentences with novel nuggets. This assumption was proven to be correct, as we were able to improve the pre-trained system's performance by 18% on the preliminary dataset.

Keywords: Information Retrieval, EPIC-QA, TAC 2020

1 Introduction

Pre-trained systems have garnered immense interest in recent years, primarily owing to their superior effectiveness compared to traditional retrieval systems, such as BM25 and Conv-KNRM [1]. Notably, this effectiveness has only been experienced on large-scale document retrieval challenges. However, the Epidemic Question Answering (EPIC-QA)

track of TAC 2020 was a sentence retrieval problem. In more detail, participants were provided with a set of 30 test queries and tasked with retrieving the top 1000 most relevant sentences [2].

Importantly, the ground truth file was created by associating each test query with a list of relevant atomic facts known as nuggets. For example, the query "Coronavirus origin" may have correlated to the following nugget: "Wuhan China". Sentences were then labelled as relevant to a specific test query only if they contained at least one nugget. To measure performance, the organizers defined a new metric and labelled it as Normalized Discounted Novelty Score (NDNS). In essence, the retrieval systems performance was largely determined by the number of unique nuggets that were present in the list of retrieved sentences.

In this work, we argue that retrieving sentences negatively impacts the performance of pre-trained systems, as the amount of available context is vastly reduced. Consequently, our aim is to improve the performance of pre-trained systems in sentence retrieval. This is significant since highly accurate sentence retrieval has the potential to improve many Natural Language Processing tasks, including question answering, summarization and novelty detection [3]. Additionally, industry leaders like Google have started incorporating sentence retrieval into their products. For example, Google Search utilizes Featured Snippets to identify and highlight the relevant sentence or span of text that satisfies a user's query [4].

Indeed, pre-trained systems have surpassed human baselines in question answering by providing sentence-level answers [5]. Consequently, one may assume that this effectiveness transfers to large-scale sentence retrieval, thus leaving little room for improvement. However, these two tasks are fundamentally different and therefore require different solutions. Briefly, in question answering, the model is provided with a handful of sentences (in the form of a passage) and asked to identify the relevant sentence for a given question. Whereas in EPIC-QA, the model is provided with a collection of more than 14 million sentences and asked to identify the top 1000 most relevant sentences.

The rest of this paper is organized as follows. Section 2 describes our proposed solution and the experiments that were used to analyze its effectiveness. Section 3 summarizes the results of our experiments, and finally, section 4 provides concluding remarks on our work.

2 Methodology

In large-scale retrieval, pre-trained systems do not examine the entire collection since this would result in excessive computational overheads. Instead, non-neural systems are used to retrieve the top k sentences for a given query. The pre-trained system then ranks these sentences in descending order of relevance. This type of staged retrieval is known as the

cascade framework and represents the most common way of applying pre-trained systems to large collections (see Figure 1) [6].

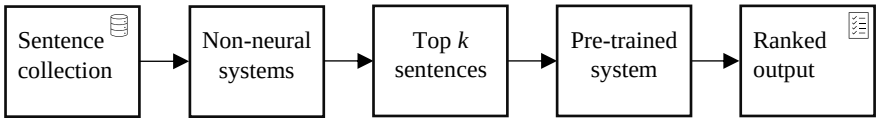

Fig. 1. Traditional cascade framework with standard data pipeline

Due to the sequenced nature of retrieval and ranking, one can argue that they are mutually exclusive events. Hence, we hypothesize that by improving the non-neural system's performance, one can subsequently improve the pre-trained system's performance. To test our hypothesis, we propose the data pipeline shown in Figure 2. As illustrated, the original sentence collection was concatenated to form a passage collection. These passages were indexed and retrieved by a selection of non-neural systems. The output of these systems was merged and subsequently split back into sentences before the ranking stage. Notably, the only difference between our data pipeline and that shown in Figure 1, is that the non-neural systems retrieve passages instead of sentences.

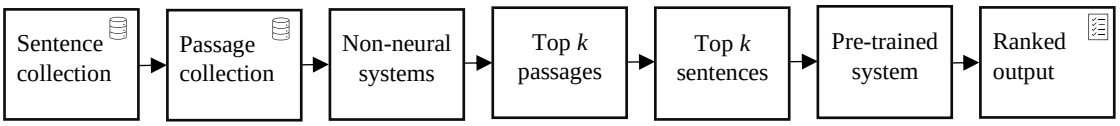

Fig. 2. Traditional cascade framework with our data pipeline

It is important to recognize that our solution does not adapt the architecture of a pre-trained system to sentence retrieval. Instead, we focused on improving the data pipeline within the cascade framework. This allowed us to provide a solution that can be applied to any pre-trained architecture.

2.1 Hypothesis Testing

Our solution assumes a strong correlation between the performance of pre-trained and non-neural systems in the cascade framework. To validate this hypothesis, we turn our attention to round 1 of the TREC-COVID challenge [7]. In more detail, we evaluate the pre-trained system's performance after ranking the output of different non-neural systems. Thus, our hypothesis is validated if peak performance is achieved when ranking the output of the highest performing non-neural system.

Further, we claimed that retrieving passages would improve the non-neural system's performance and subsequently improve the pre-trained system's performance. To understand the reasons behind this performance gain, we compare the ranked outputs between the standard pipeline, shown in Figure 1, and our pipeline, shown in Figure 2. Specifically, we compare the number of unique nuggets and relevant sentences retrieved by both pipelines.

It is worth emphasizing that NDNS is primarily determined by the number of unique nuggets in the results list. This means that the number of relevant retrieved sentences has little influence on performance. Consequently, to better understand the performance gain, we consider the metric counterpart of NDNS, which is Normalized Discounted Cumulative Gain (NDCG). In essence, NDCG evaluates a retrieval system's performance by penalizing the score every time a non-relevant sentence appears higher up in the rank than a relevant sentence.

2.2 Models

This paper utilized various non-neural systems for retrieval. Specifically, our selection included BM25 [8], ColBERT [9], DeepCT [10] and docTTTTTquery [11]. These systems were chosen based on their high performance in popular retrieval challenges such as MS MARCO. To begin with, we turn our attention to Anserini BM25. In essence, BM25 creates an inverted index that consists of term weights for each document in the collection. These weights indicate the topic of a particular document. For instance, in our paper, 'pre-trained' and 'system' may receive the highest term weights. BM25 can then measure relevance by matching query terms to documents in which those terms have the highest weights.

Each system in our selection, apart from BM25, was trained on the MS MARCO dataset. Although vastly different to EPIC-QA data, recent work suggests that MS MARCO represents an optimal training set for this challenge [12]. Notably, however, each system utilized a different training methodology that suited their functionality. Both ColBERT and DeepCT compute document representations offline using contextualized embeddings from the BERT architecture. However, DeepCT converts these document representations to term weights, while ColBERT stores them as a bag of words embedding. This means that DeepCT relies on BM25 to perform retrieval, whereas ColBERT relies on vector similarity between the query and document embeddings.

On the other hand, docTTTTTquery uses Google's T5 transformer to expand documents in the collection. This expansion consists of generated questions that the document might be able to answer. BM25 is then used to index the expanded document collection and subsequently perform retrieval. Lastly, we turn our attention to the pre-trained BERT Large architecture. In more detail, BERT represents a general-purpose language model that has achieved state-of-the-art results in Information Retrieval. Hence, it formed the basis of our pre-trained ranking model. As described by the authors in [13], the input to

BERT was formed by concatenating the query and sentence into a sequence. BERT then computes the probability of the sentence being relevant to the query, i.e., relevance score. Sentences are then sorted in descending order of relevance.

3 Results

3.1 Preliminary Dataset

In this section, we detail the experiments that were performed on the preliminary and TREC-COVID datasets. Firstly, we test the correlation between the performance of pre-trained and non-neural systems in the cascade framework. The dataset used for this experiment was sourced from round 1 of the TREC-COVID challenge. We selected TREC-COVID since it was orders of magnitude smaller than the preliminary EPIC-QA dataset, hence vastly reduced computational costs. The collection consisted of approximately 50 000 documents formed by concatenating the title and abstract fields of the metadata file.

For this experiment, the chosen dataset did not matter since our hypothesis made a general assumption about pre-trained systems in the cascade framework. Figure 3 plots the non-neural system's performance on the x-axis. The corresponding y-value represents the pre-trained system's performance after ranking the non-neural system's output. Based on the observed trend, one can conclude that a strong correlation exists between the non-neural and pre-trained system's performance.

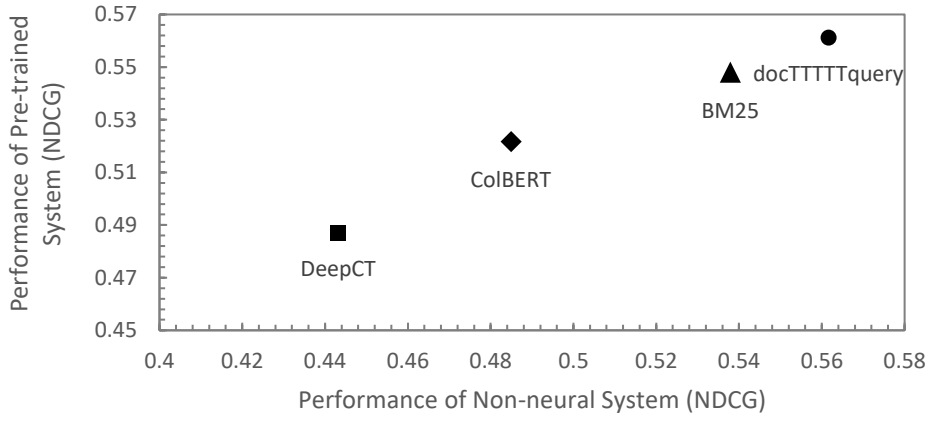

Fig. 3. Correlation between the performance of pre-trained and non-neural systems in the cascade framework

Secondly, we compare the performance of the standard pipeline to our pipeline (see Table 1). For this experiment, we utilized the preliminary EPIC-QA dataset, which consisted of approximately 14 million sentences. Due to high computational overheads, our selection of non-neural systems was limited to BM25 and ColBERT. Nevertheless, the results of this experiment are valid since both the standard pipeline and our pipeline utilized identical non-neural systems.

Table 1. NDNS-Exact and NDCG performance of BERT using the standard pipeline and our pipeline

Data Pipeline	BERT Performance (NDNS-Exact)	BERT Performance (NDCG)
Standard Pipeline	0.1564	0.0832
Our Pipeline	0.1855	0.1139

Lastly, we analyze how the additional context of passages helped our pipeline outperform the standard pipeline, as evident in Table 1. To begin with, Figure 4 compares the nugget recall of both pipelines for each test query. Essentially, a nugget recall value of 0.7 implies that 70% of all unique nuggets are present in the output. By observing the trend in Figure 5, we see a strong correlation between nugget recall and performance. In more detail, Figure 5 plots the difference in performance against the difference in nugget recall between our pipeline and the standard pipeline for each test query.

Similarly, we also compare the relevant sentence recall of each pipeline, as shown in Figure 6. A sentence was considered relevant only if it contained at least one nugget. Figure 7 illustrates the correlation between the difference in NDCG performance and sentence recall of our pipeline and the standard pipeline.

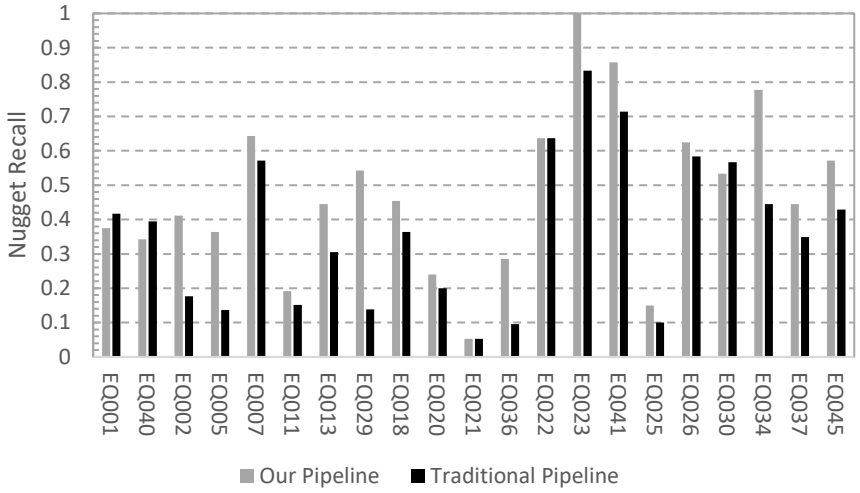

Fig. 4. Nugget recall

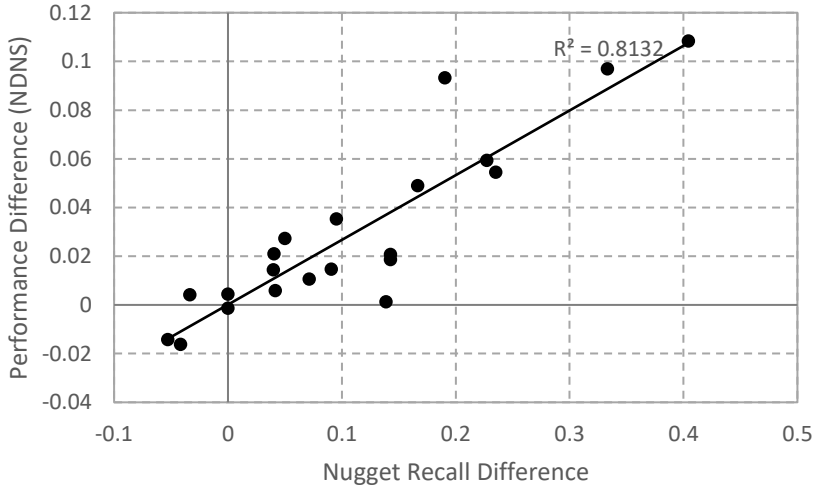

Fig. 5. Performance vs nugget recall

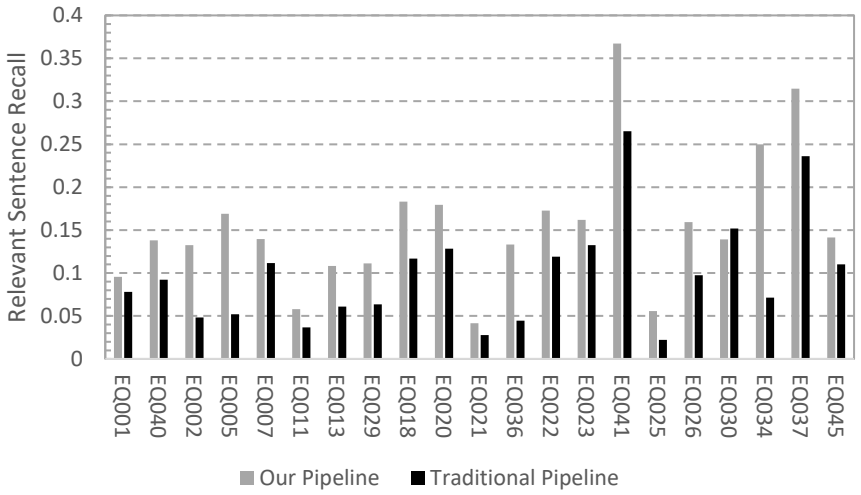

Fig. 6. Sentence Recall

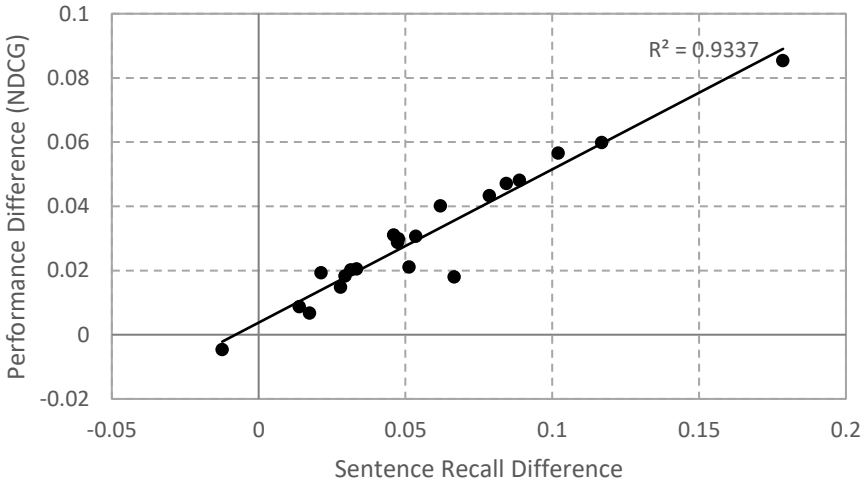

Fig. 7. Performance vs. sentence recall

3.2 Primary Dataset

After validating the effectiveness of our approach on the preliminary dataset, we applied it to Task A of the primary round. It is worth emphasizing that our experiments showed a strong correlation between the non-neural system's recall and the pre-trained system's performance. As a result, we utilized our entire selection of non-neural systems for the primary round, i.e., BM25, ColBERT, DeepCT and docTTTTTquery. It was assumed that recall (and performance) could be further increased by merging the output of multiple non-neural systems. The NDNS performance of BERT after ranking the output of these systems is summarized in Table 2. Importantly, these performances correspond to Run_1.

Table 2. NDNS performance of BERT on the primary dataset

NDNS	Value
Partial	0.3613
Relaxed	0.3624
Exact	0.4101

4 Conclusion

This work aimed to improve the performance of pre-trained systems in sentence retrieval. Based on our experiments, we realized a strong correlation between the non-neural and pre-trained system's performance in the cascade framework. Hence, although counterintuitive, we achieved our aim by improving the non-neural system's performance. Specifically, we proposed that the non-neural system should retrieve passages instead of sentences. It was found that the additional context of passages improved the non-neural system's performance. Further analysis showed that the observed performance gain was highly correlated to an increase in recall.

References

[1] N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and E. M. Voorhees, 'Overview of the TREC 2019 deep learning track', *arXiv:2003.07820 [cs]*, Mar. 2020, Accessed: May 17, 2020. [Online]. Available: http://arxiv.org/abs/2003.07820.

[2] Travis R. Goodwin *et al.*, 'Overview of the 2020 Epidemic Question Answering Track', vol. 13.

[3] Vanessa Graham Murdock, 'Aspects of sentence retrieval', University of Massachusetts, Amherst, 2006.

[4] A. Strzelecki and P. Rutecka, *Featured Snippets Results in Google Web Search: An Exploratory Study*. 2019.

[5] T. Kwiatkowski *et al.*, 'Natural Questions: A Benchmark for Question Answering Research', *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453–466, Aug. 2019, doi: 10.1162/tacl_a_00276.

[6] L. Wang, J. Lin, and D. Metzler, 'A Cascade Ranking Model for Efficient Ranked Retrieval', in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2011, pp. 105–114, doi: 10.1145/2009916.2009934.

[7] K. Roberts *et al.*, 'TREC-COVID: Rationale and Structure of an Information Retrieval Shared Task for COVID-19', *J Am Med Inform Assoc*, May 2020, doi: 10.1093/jamia/ocaa091.

[8] P. Yang, H. Fang, and J. Lin, 'Anserini: Enabling the Use of Lucene for Information Retrieval Research', Aug. 2017, pp. 1253–1256, doi: 10.1145/3077136.3080721.

[9] O. Khattab and M. Zaharia, 'ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT', *arXiv:2004.12832 [cs]*, Jun. 2020, Accessed: Apr. 01, 2021. [Online]. Available: http://arxiv.org/abs/2004.12832.

[10] Z. Dai and J. Callan, 'Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval', *arXiv:1910.10687 [cs]*, Nov. 2019, Accessed: Apr. 01, 2021. [Online]. Available: http://arxiv.org/abs/1910.10687.

[11] R. Nogueira and J. Lin, 'From doc2query to docTTTTTquery', p. 3.

[12] Y. Rughbeer, A. W. Pillay, and E. Jembere, 'Dataset Selection for Transfer Learning in Information Retrieval', in *Artificial Intelligence Research*, Cham, 2020, pp. 53–65, doi: 10.1007/978-3-030-66151-9_4.

[13] R. Nogueira and K. Cho, 'Passage Re-ranking with BERT', *arXiv:1901.04085 [cs]*, Feb. 2019, Accessed: Apr. 14, 2020. [Online]. Available: http://arxiv.org/abs/1901.04085.