York University at TREC 2021: Deep Learning Track

Yizheng Huang and Jimmy Huang
Information Retrieval and Knowledge Management Research Lab
York University, Toronto, Canada
{hyz, jhuang}@yorku.ca

Abstract

This is our first time to participate in TREC Deep Learning track. We submitted three BERT-based runs "YorkU21a", "YorkU21b" and "YorkU21c", where YorkU21a has the best performance. Our main goal is to explore the possibility of combining deep learning with traditional BM25 retrieval method. In this paper, we discuss the results of using the summation method to combine the above two approaches and provide some illustrative analyses on the impact of different retrieval strategies on the results.

Keywords

BM25, Deep Learning, Meta Search

1 Introduction

Information retrieval (IR) systems search for the relevant information from a collection of data based on various retrieval models such as probabilistic models, language models and deep learning models. The meta-search system [1] will access multiple IR systems simultaneously and gather their ranking results into a single ranked output. Since the IR systems retrieve different results, the metasearch system provides a way to combine the results of multiple systems to make the best match.

Deep learning models like Transformers [2] and Bert [3], calculate the similarity between word embeddings by semantic search to rank passages, which is a semantic-based ambiguous matching. Although BM25 [4, 5] is keyword-based and cannot understand context and distinguish synonyms in the traditional way of retrieval, it is an exact match. IRLab at York University participated in the TREC 2021 Deep Learning Track for passage (full) ranking task. Our primary goal was to explore the results of combining the above two different search methods.

This paper is organized as follows: Section 2 presents our methods applied in passage ranking. Section 3 details our experimental results on different evaluation metrics. Finally, in Section 4, we discuss the conclude and present our future work.

2 Our Methods

We have a three-stage framework. In the first stage, we did not use any dataset to train a new model because our focus is to combine deep learning and BM25. So, we utilized the Sentence-Bert trained by UKPLab [6]. We utilized msmarco-MiniLM-L-6-v3 for the normalized embedding and retrieving a candidate set because it has a faster computation speed. The candidate set was a submitted run as YorkU21b. We encoded each search query as a sentence embedding and used semantic search to calculate its relevance to the embeddings of the dataset. This would retrieve the most relevant 100 hits from each jsonl file, instead of retrieving them from the entire dataset. And we utilized two pre-trained cross-coder model ms-marco-MiniLM-L-12-v2 and ms-marco-MiniLM-L-6-v2 to re-rank the candidate set. The results of the above three rankings were voted, and the top 100 most relevant passages for each query were selected as the final result which was our submitted run of YorkU21a. For comparison, we also submitted the result YorkU21c, which was re-ranked only by ms-marco-MiniLM-L-6-v2.

During the second stage, we used the Anserini [7] with default parameters to obtain the result of BM25. Finally, we used a method to combine the result of YorkU21a and BM25.

2.1 Combine Method

The way we considered combining BM25 and Deep Learning is straightforward, combining the two results in the simplest way. Some simple combination methods are introduced by Fox and Shaw [8], such as CombMax, CombMin, CombSUM, and CombMed, and CombSUM is the best performing combination method. In this paper, we used CombSUM as a method to combine the results of BM25 and Deep Learning.

In this paper, we will use the summation method to combine the two retrieval results as follows:

$$CombSUM = \sum_{p \in P} S(q) \tag{1}$$

where P represents a set of retrieved passages and p represents each passage, q is a query and $S(q)$ is the score of the passage retrieved by the query.

2.2 Normalization Method

Since the use of different retrieval systems will have different weighting methods and thus produce quite different ranges of similarity values, it is necessary to apply normalization methods to the different retrieval results. We used three different methods for normalization, which are Rescaling, Mean Normalization and Z-score Normalization.

Rescaling, also known as min-max normalization, is the simplest way to reduce the range of data to [0, 1] or [-1, 1], where the general formula for normalization of [0, 1] is as follows:

$$s' = \frac{s - \min(s)}{\max(s) - \min(s)} \tag{2}$$

where s is an original score of ranking, s' is the normalized ranking score.

Mean Normalization is another way to normalize data, it works by calculating and subtracting the mean value of each data. A common practice is to divide this value by the range or standard deviation, and the formula is as follows:

$$s' = \frac{s - \bar{s}}{\max(s) - \min(s)} \tag{3}$$

where $\bar{s}$ is the mean of scores.

When the same process is done using standard deviations as denominators, the process is called Standardization, also known as Z-score Normalization. Z-score Normalization is widely used method for normalization in many machine learning algorithms. This method gives the mean and standard deviation of the original data to standardize the data. The processed data conforms to the standard normal distribution, that is, the mean is 0 and the standard deviation is 1, and the formula is as follows:

$$s' = \frac{s - \mu}{\sigma} \tag{4}$$

where μ is the mean of all sample data and σ is the standard deviation of all sample data.

3 Results

Since YorkU21a achieves the best result for deep learning retrieval, we only use this run in combination with BM25. Our experiment runs are denoted as: YorkU21a, Anserini_BM25, CombSUM_rescal, CombSUM_zscore, and CombSUM_mean, where YorkU21a is our official submission and others are unofficial. Table 1 presents the detailed descriptions of these runs are as follows.

Table 1: Runs Description

Runs	**Description**
Anserini_BM25	Use BM25 algorithm to obtain a baseline run by Anserini.
YorkU21a	Use Sentence-Bert to obtain the best re-rank result.
CombSUM_rescal	Use CombSUM method to combine the above two methods by Rescaling.
CombSUM_zscore	Use CombSUM method to combine the above two methods by Z-score Normalization.
CombSUM_mean	Use CombSUM method to combine the above two methods by Mean Normalization.

Table 2: Results with Different Method

Runs	**MAP**	**P@10**	**NDCG@10**	**NDCG@100**
Anserini_BM25	0.1357	0.3547	0.4458	0.3913
YorkU21a	0.5308	0.6151	0.6965	0.5779
CombSUM_rescal	0.2917	0.5491	0.6370	0.5654
CombSUM_mean	0.2230	0.5283	0.6122	0.4691
CombSUM_zscore	0.2602	0.4906	0.5875	0.5379

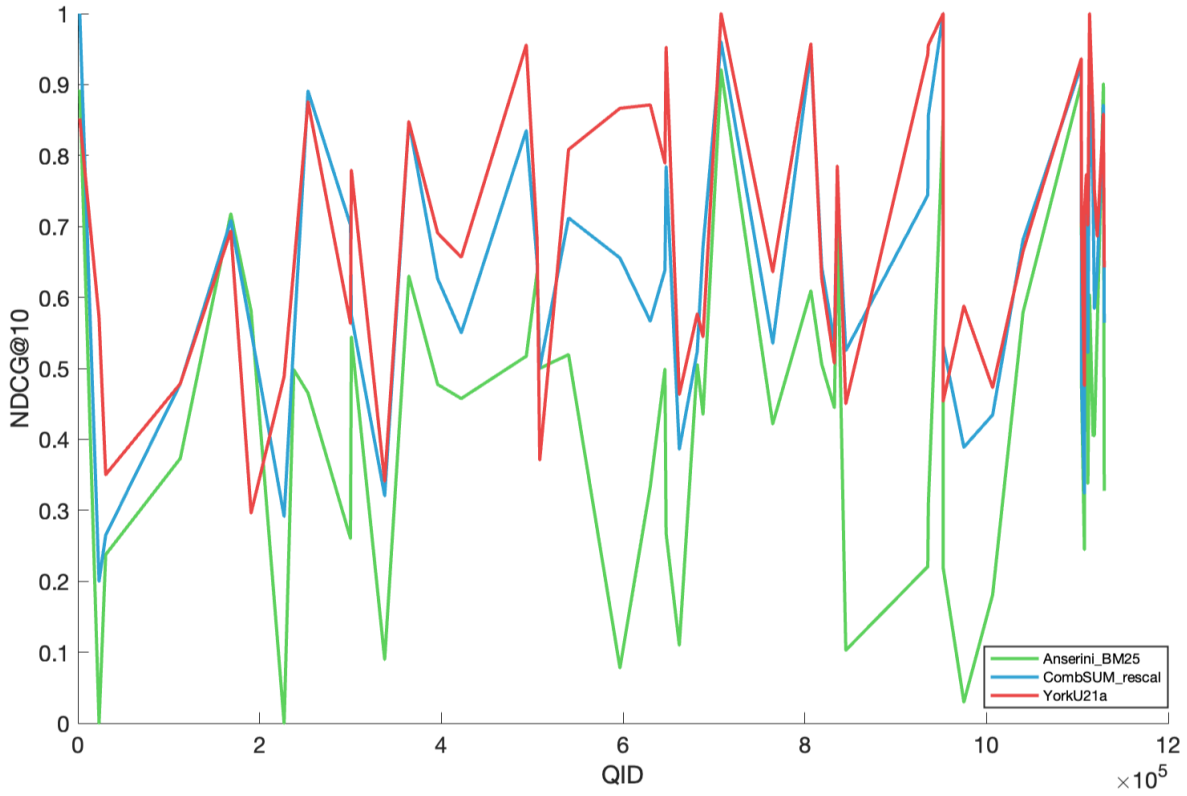

Figure 1: Comparison of Three Methods

Results of our full passage ranking runs are shown in Table 2. NIST evaluation provided a four-degree level: '0' irrelevant, '1' related, '2' highly relevant, and '3' perfectly relevant. For passages, a judgment level of '1' means the passage was related to the query but didn't actually answer it, so for measures that use binary relevance judgments a '1' is not relevant. Except for the calculation of NDCG which needs to use all the levels, the rest of the evaluation calculation should not consider level '1'.

We can clearly find that the deep learning model has the best result. And different normalization methods have an effect on the results, where Rescaling achieves the best performance. To better understand the differences between the deep learning, BM25 and combined methods, we compare them together as shown in the Figure 1. It can be seen that when the performance difference between deep learning and BM25 is large, the combined result becomes worse than deep learning, and conversely, if the performance of both is close, the result will be slightly better than deep learning. The interesting thing, however, is that in the overall case where deep learning is much stronger than BM25, BM25 still performs better in some queries. We analyzed one of the particular cases, as shown in Table 3.

In Table 3, we can find that the keyword-based BM25 model is more likely to find relevant passages when the query is a short sentence, while the semantic-based deep learning model will find passages that partially match the keywords rather than the exact matches. In other cases, we found that BM25 performs poorly when the query is a long sentence, as it is an exact match, and deep learning shows strong capabilities.

4 Discusstions

The performance of deep learning-based retrieval is much better than that of BM25-based retrieval. If the performance gap between the two is too large for the query, combining the two results will instead yield a worse result. But when the performance of them is close, combining the two results will produce a better result.

From the retrieval results, the traditional BM25 is more suitable for retrieving queries with short sentences, while deep learning has better performance in retrieving long sentences based on semantics. However, the retrieval performance of deep learning under short sentences is sometimes far inferior to that of BM25. For example, when matching people's names, the deep learning model tends to match passages that contain partial information about the person's name, while BM25 will precisely match passages with the exact same person's name, which will lead to inaccuracy of deep learning.

For our future work, we will first focus on combining the advantages of traditional models for keyword-based exact matching with deep learning models to further improve the search performance. Second, we will apply our methods for more applications (such as biomedicine and Clinical IR) [9, 10, 11].

Table 3: Comparison of Deep Learning (NDCG@10: 0.2962) and BM25 (NDCG@10: 0.5815) in Query 190623: "for what is david w. taylor known"

Runs	Rank	Related	Passage
YorkU21a	1	3	Rear Adm. David W. Taylor. Rear Admiral David Watson Taylor, USN (March 4, 1864 - July 28, 1940) was a naval architect and engineer of the United States Navy. He served during World War I as Chief Constructor of the Navy, and Chief of the Bureau of Construction and Repair. Taylor is best known as the man who constructed the first experimental towing tank ever built in the United States.
YorkU21a	2	0	World Champ David taylor the magic man. World Champ. David taylor the magic man. David Taylor, widely known as The Magic Man, is a 4x NCAA All-American, 4x BIG 10 Champion, and a 2x NCAA Champion – and he just getting started. Having wrapped up his NCAA career in March of 2014, David is just getting started on his international career and ultimately, his quest for Gold in Tokyo, 2020.
YorkU21a	3	0	Taylor is best known for his contributions to microhistory, exemplified in his William Cooper's Town: Power and Persuasion on the Frontier of the Early American Republic (1996). Using court records, land records, letters and diaries, Taylor reconstructed the background of founder William Cooper from Burlington, New Jersey, and the economic, political and social history related to the land speculation, founding and settlement of Cooperstown, New York, after the American Revolutionary War.
Anserini_BM25	1	2	History. The facility was previously known as the David W. Taylor Naval Ship Research and Development Center; it was renamed David Taylor Research Center (DTRC) in 1987 and later became the Carderock Division of the Naval Surface Warfare Center in 1992.
Anserini_BM25	2	2	David S. Taylor, CEO of Procter and Gamble. David Taylor (Wisconsin judge), American jurist and legislator. David W. Taylor, U.S. Navy admiral and engineer. David Taylor (banker), banker. David Taylor (veterinarian), television presenter on animal subjects.
Anserini_BM25	3	3	Rear Adm. David W. Taylor. Rear Admiral David Watson Taylor, USN (March 4, 1864 - July 28, 1940) was a naval architect and engineer of the United States Navy. He served during World War I as Chief Constructor of the Navy, and Chief of the Bureau of Construction and Repair. Taylor is best known as the man who constructed the first experimental towing tank ever built in the United States.

Acknowledgement

This research is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the York Research Chairs (YRC) program.

References

[1] M. Manoj and Elizabeth Jacob. Information retrieval on internet using meta-search engines : A review. *Journal of Scientific Industrial Research*, 67(10):739–746, 2008.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

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

[4] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Mike Gatford, and A. Payne. Okapi at TREC-4. In Donna K. Harman, editor, *Proceedings of The Fourth Text REtrieval Conference, TREC 1995, Gaithersburg, Maryland, USA, November 1-3, 1995*, volume 500-236 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 1995.

[5] Ben He, Jimmy Xiangji Huang, and Xiaofeng Zhou. Modeling term proximity for probabilistic information retrieval models. *Inf. Sci.*, 181(14):3017–3031, 2011.

[6] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv:1908.10084 [cs]*, August 2019.

[7] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1253–1256, 2017.

[8] Edward A. Fox and Joseph A. Shaw. Combination of multiple searches. In *TREC-2: Text retrieval conference*, number 500215, pages 105–108, 1994.

[9] Xiangji Huang, Fuchun Peng, Dale Schuurmans, Nick Cercone, and Stephen E. Robertson. Applying machine learning to text segmentation for information retrieval. *Inf. Retr.*, 6(3-4):333–362, 2003.

[10] Xiangji Huang, Ming Zhong, and Luo Si. York university at TREC 2005: Genomics track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005, Gaithersburg, Maryland, USA, November 15-18, 2005*, volume 500-266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2005.

[11] Xiangji Huang and Qinmin Hu. A bayesian learning approach to promoting diversity in ranking for biomedical information retrieval. In James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel, editors, *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 307–314. ACM, 2009.