STIS at the NTCIR-15 Data Search Task: Document Retrieval Re-ranking

Lya Hulliyyatus Suadaa
Department of Statistical Computing
Politeknik Statistika STIS
lya@stis.ac.id

Lutfi Rahmatuti Maghfiroh
Department of Statistical Computing
Politeknik Statistika STIS
lutfirm@stis.ac.id

Isfan Nur Fauzi
Department of Statistical Computing
Politeknik Statistika STIS
isfan@stis.ac.id

Siti Mariyah
Department of Statistical Computing
Politeknik Statistika STIS
sitimariyah@stis.ac.id

ABSTRACT

The STIS team participated in the English subtasks of the NTCIR-15 Data Search Task, exploring metadata of document as document features consisting of title, description, and tags of documents. Baseline models used traditional information retrieval of Anserini for ad-hoc retrieval of governmental statistical data. The availability of query-document relevance datasets annotated by humans encourages elaborating those datasets to improve candidate retrieved documents. We proposed a re-ranking document retrieval approach using relevance level classifiers of query-document pairs to improve document retrieval performance. For top candidate documents of Anserini, we re-ranked documents by using Bi-LSTM classifier and Finetuned BERT-based classifiers. Our results show that the re-ranking approach by finetuning the BERT-based relevance level classifier improves the document retrieval quality of Anserini.

TEAM NAME

STIS

SUBTASKS

English subtask

1 INTRODUCTION

The STIS team participated in the English subtasks of the NTCIR-15 Data Search Task [4]. In this subtask, we have to generate a ranked list of statistical data sets for each query from data collections of the US government (Data.gov). Baseline models used traditional information retrieval of Anserini to retrieve relevance datasets using their metadata, such as title and description of the statistical data. There is still a gap in document retrieval performance since the baselines only count the lexical similarity between queries and documents.

The availability of query-document relevance datasets annotated by humans encourages elaborating those datasets to improve candidate retrieved documents. Human annotation data are assumed can represent the semantic similarity between queries and documents. Therefore, we proposed a re-ranking approach to improve candidate retrieved documents of Anserini by utilizing the annotated datasets through neural classifiers.

Akkalyoncu Yilmaz et al. [1] proposed a similar re-ranking approach in Birch by integrating Anserini with BERT-Ranker to select the most related sentences in the documents. Nogueira and Cho [5] also used BERT in query-based re-ranking by classifying passages into relevant and non-relevant ones and using the probability of the passage being relevant as a new score. Adopting those mechanisms, we propose a re-ranking approach by combining traditional information retrieval and neural relevance classifier scores. We implement Bi-LSTM and various BERT models in our proposed classifiers.

The remainder of the paper is organized as follows. Section 2 describes the detail of the proposed document retrieval re-ranking

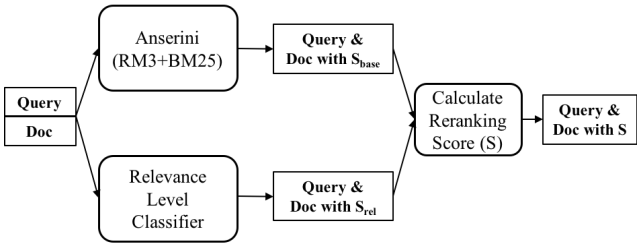

Figure 1: Re-ranking architecture.

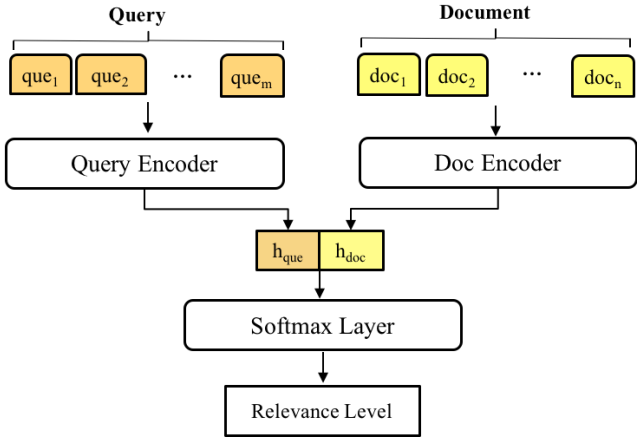

Figure 2: The architecture of our BiLSTM model to classify relevance level between query and document.

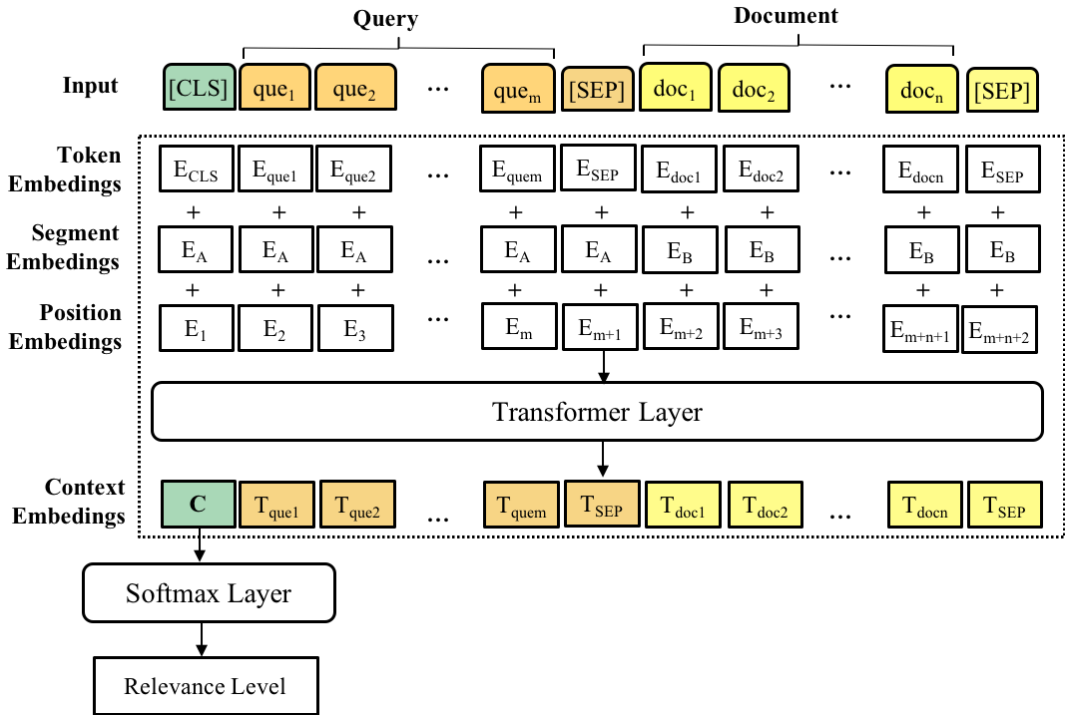

Figure 3: The architecture of our BERT-based model to classify relevance level between query and document.

approach, section 3 explains our experiments, and section 4 concludes this paper.

2 DOCUMENT RETRIEVAL RE-RANKING

The architecture of the proposed re-ranking approach can be seen in Figure 1. Following Birch [1], we first use Anserini [7] to retrieve candidate documents. We use score resulted by Anserini as base score (S_{base}). We then develop neural relevance level classifiers to obtain the relevance score (S_{rel}) of each candidate document. We have three classes of relevance level: irrelevant ($L0$), partially relevant ($L1$), and highly relevant ($L2$).

We calculate the final re-ranking score by combining the base score and relevance score as follows:

$$S = (1 - \alpha)S_{base} + \alpha S_{rel}, \tag{1}$$

where α is the weight of relevance score. We convert relevance level categories into numbers representing their relevance level as follows:

$$S_{rel} = \begin{cases} 0, & \text{if relevance level} = L0 \\ 5, & \text{if relevance level} = L1 \\ 10, & \text{if relevance level} = L2 \end{cases} \tag{2}$$

2.1 BiLSTM Relevance Classifier

We use the BiLSTM encoder to compute the context vector and use the last hidden state for context representation of query (h_{query}) and document (h_{doc}). Then, we feed the concatenation of query and document context into the softmax layer to get the relevance level probability:

$$p_{rel} = \text{softmax}(h_{que}; h_{doc}). \tag{3}$$

The architecture of our BiLSTM classifier is shown in Figure 2.

2.2 BERT-based Relevance Classifier

We finetune pre-trained encoders BERT and roBERTa for our classification task. Adopting the finetuned BERT approach in question answering task [2], we preprocess query and document token as input by inserting two special tokens, [CLS] and [SEP]. The [CLS] token is added to the beginning of input, and the [SEP] token is inserted after the query token to separate the query and document segments.

As shown in Figure 3, we denote the input embedding as E and use the token representations from the top hidden layers as context embeddings. We feed the first context C as a representation of the input sequence to the softmax layer to obtain the relevance level probability:

$$p_{rel} = \text{softmax}(C). \tag{4}$$

3 EXPERIMENTS

3.1 Implementation Details

We used metadata of document as document features consisting of title, description and tags of documents. We first retrieve the top hundred documents for each query by using Anserini. Following Birch [1], we choose RM3+BM25 model to get the base score. Then, we train the candidate documents accompanied by the relevance

Relevance Classifier	Doc Feature	P_{L0}	R_{L0}	F_{L0}	P_{L1}	R_{L1}	F_{L1}	P_{L2}	R_{L2}	F_{L2}	P	R	F
BiLSTM	title + desc	0.94	0.65	0.77	0.12	0.54	0.19	0	0	0	0.8664	0.6341	0.7148
BiLSTM	title + tag	0.96	0.46	0.62	0.11	0.76	0.19	0	0	0	0.886	0.4822	0.5847
Finetuned BERT-base	title + desc	0.97	0.70	0.82	0.18	0.75	0.30	0	0	0	0.9051	0.7039	0.771
Finetuned BERT-base	title + tag	0.98	0.73	0.84	0.21	0.85	0.34	0	0	0	0.9133	0.7376	0.794
Finetuned BERT-large	title + desc	0.99	0.64	0.78	0.18	0.91	0.30	0	0	0	0.916	0.6595	0.735
Finetuned BERT-large	title + tag	0.99	0.57	0.73	0.16	0.93	0.27	0	0	0	0.9167	0.5982	0.6847
Finetuned roBERTa-base	title + desc	0.98	0.42	0.59	0.12	0.89	0.21	0	0	0	0.9085	0.4572	0.5569
Finetuned roBERTa-base	title + tag	0.98	0.61	0.75	0.15	0.83	0.26	0	0	0	0.9038	0.6191	0.7031

Table 1: Performance of relevance level classifier.

Anserini	Relevance Classifier	Doc Feature	nDCG@3	nDCG@5	nDCG@10	nERR@3	nERR@5	nERR@10	Q-mea
RM3+BM25	-	title + desc	0.195	0.202	0.213	0.230	0.201	0.215	0.228
RM3+BM25	BiLSTM	title + desc	0.165	0.175	0.197	0.187	0.182	0.194	0.211
RM3+BM25	BiLSTM	title + tag	0.167	0.163	0.172	0.164	0.186	0.192	0.201
RM3+BM25	Finetuned BERT-base	title + desc	**0.201**	0.201	**0.221**	0.199	**0.227**	**0.234**	**0.249**
RM3+BM25	Finetuned BERT-base	title + tag	**0.230**	**0.228**	**0.237**	0.217	**0.248**	**0.255**	**0.264**
RM3+BM25	Finetuned BERT-large	title + desc	0.189	0.195	0.211	0.202	**0.202**	0.214	0.226
RM3+BM25	Finetuned BERT-large	title + tag	0.172	0.171	0.192	0.185	0.190	0.199	0.212
RM3+BM25	Finetuned roBERTa-base	title + desc	0.155	0.151	0.177	0.171	0.175	0.181	0.198
RM3+BM25	Finetuned roBERTa-base	title + tag	0.165	0.175	0.197	0.187	0.182	0.194	0.211

Table 2: Results from NTCIR-15 Data Search. Values in bold outperform the baseline. There are no significant differences between run pairs.

level of each pair of query and document in relevance level classifiers. We implement our classifiers using the AllenNLP library [3]. In the query and document encoders of our BiLSTM classifier, we use two-layer BiLSTMs with 128 hidden sizes, initialized by GloVe [6]. For optimization in the training phase, we use Adam as the optimizer with a batch size of 5 and a learning rate of 3×10^{-3} and 3×10^{-5} in our BiLSTM and BERT-based model, respectively. We split the original train set into a 90% train set and a 10% validation set. We then trained the model for a maximum of ten epochs with early stopping on the validation set (patience of 5). We give more weight to our relevance score by setting α to 0.7.

3.2 Results

We evaluated our relevance level classifiers as shown in Table 1. First, we matched our candidate documents from Anserini with query-document relevance testing datasets released by the organizer. Using RM3+BM25 model, we have 51.25% pairs of query and document match with testing dataset (average of 33.66% documents per query). We have 91.37% support data of L0 class, 8% of L1 class, and 0.6% for L2 class. Since we have imbalance datasets, we also present precision, recall, and F1 score for each relevance level.

As shown in Table 1, the precision of Finetuned BERT-based classifiers mostly scored better than the BiLSTM classifier. It means that incorporating pre-trained encoders lead to better accuracy than the BiLSTM encoder model. However, the Finetuned BERT-based using roBERTa drop the F1 score.

Comparing the performance of each relevance level, the F1 scores of L1 are below 35%, and there is no true prediction for all L2 classes. A handling imbalance dataset is recommended for future research.

The overall performances of our re-ranking approach are shown in Table 2. We can see that our re-ranking mechanism using the Finetuned BERT-base classifier outperformed the baseline (without re-rank). However, the performances drop when the candidate documents were re-ranked by other classifiers.

We also compare several document features as inputs, a combination of title and description, and a combination of title and tags. From Table 1 and Table 2, since the results are not consistent between models, we still can not conclude whether using title and tags feature is better than only using title and description.

4 CONCLUSIONS

In this paper, we proposed a re-ranking approach for ad-hoc retrieval of governmental statistical data. We used metadata of document as document features consisting of title, description and tags of documents. We re-ranked candidate documents for each query from Anserini by using relevance level classifiers of query-document pairs. We proposed a Bi-LSTM classifier and Finetuned BERT-based classifiers, and presented the experimental results.

REFERENCES

[1] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Applying BERT to Document Retrieval with Birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations.* Association for Computational Linguistics, Hong Kong, China, 19–24. https://doi.org/10.18653/v1/D19-3004

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* Association for Computational Linguistics, Minneapolis, Minnesota,

4171–4186. https://doi.org/10.18653/v1/N19-1423

[3] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. Association for Computational Linguistics, Melbourne, Australia, 1–6. https://doi.org/10.18653/v1/W18-2501

[4] Makoto P. Kato, Hiroaki Ohshima, Ying-Hsang Liu, and Hsin-Liang Chen. 2020. Overview of the NTCIR-15 Data Search Task. In *Proceedings of the NTCIR-15 Conference*.

[5] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR* abs/1901.04085 (2019). arXiv:1901.04085 http://arxiv.org/abs/1901.04085

[6] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[7] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. 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, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 1253–1256. https://doi.org/10.1145/3077136.3080721