SLWWW at the NTCIR-16 WWW-4 Task

Yuya Ubukata
Waseda University
911_yub@toki.waseda.jp

Masaki Muraoka
Waseda University
muraokamasaki@suou.waseda.jp

Sijie Tao
Waseda University
tsjmailbox@ruri.waseda.jp

Tetsuya Sakai
Waseda University
tetsuya@waseda.jp

ABSTRACT

The SLWWW team participated in the NTCIR-16 We Want Web with CENTRE (WWW-4) task. This paper reports our approach and results in the ad hoc web search task. We applied two different methods to generate NEW runs, COIL (Contextualized Inverted List) and PARADE (Passage Representation Aggregation for Document Reranking). We also tried to reproduce the KASYS run which was a top-performing run in the WWW-3 task. Furthermore, we conducted a per-topic analysis for a more in-depth discussion.

TEAM NAME

SLWWW

SUBTASKS

English

1 INTRODUCTION

The SLWWW team participated in the NTCIR-16 We Want Web with CENTRE (WWW-4) task [10]. This paper reports our approach for generating NEW runs and REP run in the ad hoc web search task.

We used pretrained transformer models to generate two different approaches for the NEW runs. One approach called COIL was proposed by Gao et al. [5], which stores representation vectors produced by BERT [4] into inverted lists to perform the contextualized exact match. Another approach called PARADE was proposed by Li et al. [6], which produces a document relevance score by aggregating passage-level representations directly into an overall document-level representation. We also tried to reproduce the KASYS run [12] in the WWW-3 task [11], which aggregates the sentence relevance score produced by BERT and the original document score to generate the final document score.

The remainder of this paper is organized as follows. Section 2 describes the details of our approach, and Section 3 reports the evaluation results and a per-topic analysis. Section 4 concludes this paper with some future work.

2 RETRIEVAL APPROACHES

In this section, we describe the four NEW runs and a REP run submitted to the task. Among the four NEW runs, three are variants of the COIL system, and the rest is a PARADE system.

2.1 Model

2.1.1 COIL. In recent years, information retrieval systems have shifted from methods using exact lexical matching such as BM25 to methods using pretrained transformer models such as BERT to perform soft semantic matching. In exact lexical matching methods, the search efficiency of inverted index-based retrieval is traded off for the inability to consider contextual information, resulting in vocabulary mismatch and semantic mismatch. On the other hand, soft matching approaches, while capable of handling mismatch problems, lose computational efficiency.

In order to incorporate the best of both approaches, Gao et al. [5] proposed COIL[1], which stores representation vectors into inverted lists to perform contextualized exact matching. In detail, we first encode tokens in the documents to compute contextualized vector representations using BERT. The vector representations are then stored into the corresponding token's inverted index along with the document id. When the query comes in, COIL computes contextualized vector representations of tokens in the query, refers to the inverted index of the corresponding token, and calculates the score. Furthermore, the special [CLS] token is used to mitigate the problem of vocabulary mismatch.

COIL has succeeded in outperforming classical lexical retrievers and the state-of-the-art deep language model retrievers and reducing latency. However, there is a problem that it cannot handle documents that exceeds the 512 tokens input limit of BERT. To deal with this problem, we split long documents into chunks of 510 tokens before encoding. Once every chunks are encoded, chunks are attached back together to form the original document. After this process, the representation of all the tokens in the document can be stored into an inverted index.

2.1.2 PARADE. For the task of document ranking, various methods using pretrained transformer models have emerged and shown their effectiveness [7]. Due to the input length limit of these models, long documents need to be broken down into passages and encoded. Several approaches have been proposed to aggregate the passage-level signals, including a method that aggregate the passage scores to obtain a document score. Li et al. [6] proposed PARADE, which produces a document relevance score by aggregating the passage-level representations directly into an overall document-level representation.

In PARADE, documents are split into a fixed number of passages. When there are fewer passages, the passages are padded and later masked out. When there are more passages, the first and the last passages are kept, and the remaining passages are randomly chosen. A passage representation is computed for each passage query pair by a pretrained transformer encoder. Of the six different approaches proposed by the authors to summarize the passage

[1] https://github.com/luyug/COIL

Table 1: Summary of our submitted runs

Run name	Method	Description
SLWWW-CO-REP-1	KASYS	Reproduction of KASYS-E-CO-NEW-1 at the NTCIR-15 WWW-3 task
SLWWW-CO-NEW-2	COIL	The input documents are divided into chunks of 510 tokens to fit in BERT. Retrieval was performed on corpus A.
SLWWW-CO-NEW-3	COIL	The input documents are divided into chunks of 510 tokens to fit in BERT. Retrieval was performed on corpus B.
SLWWW-CO-NEW-4	COIL	The input documents are truncated to 512 tokens. Retrieval was performed on corpus A.
SLWWW-CO-NEW-5	PARADE	Rerank the top 1,000 documents retrieved by BM25. PARADE-Transformer is used for passage representation aggregation.

relevance representations into a single dense representation of a document, we adopted PARADE-Transformer. In this method, the [CLS] token and all the passage representations are concatenated and fed to the transformer encoders. The [CLS] vector output from the last encoder is treated as the relevance representation between the query and the whole document.

2.1.3 REP run. The target run of the WWW-4 REP run is the KASYS run (KASYS-E-CO-NEW-1) submitted to the NTCIR-15 WWW-3 task. Their run was based on the sentence score aggregation approach, which aggregates the top sentence relevance score and the original document score to generate the final document score.

2.2 Experimental Setup

We submitted five runs to the WWW-4 task as shown in Table 1. For all runs, we used the WWW-3 dataset, which is the Clueweb12-B13[2], as a validation dataset.

2.2.1 COIL. Following Gao et al.'s work [5], we used BERT-base (uncased, 768 CLS dimension, 110M parameters) as a pretrained language model. We trained the model with MSMARCO passage dataset [2], which is consisted of user queries obtained from Bing's search logs and passages extracted from web documents. Since it was not practical to perform retrieval from the full Chuweb21 Corpus due to computational efficiency, we created two subsets of the corpus, corpus A and corpus B. Corpus A is a compilation of the top 1,000 most relevant documents for each topic, extracted using BM25 implemented in Anserini [14]. Corpus B is a larger version of corpus A, which is a compilation of the top 10,000 relevant documents for each topic. Both corpora exclude duplicates, with corpus A containing 49,139 documents and corpus B about 478,529 documents. For SLWWW-CO-NEW-2 and SLWWW-CO-NEW-3, we divided the long input documents into chunks of 510 tokens to fit in BERT. For SLWWW-CO-NEW-4, documents are truncated to the first 512 tokens following the original work.

2.2.2 PARADE. Following Lin et al.'s work [6], we first retrieved the top 1,000 relevant documents for each topic, extracted using BM25 implemented in Anserini [14]. We then used the ELECTRA [3] model fine-tuned on the MSMARCO passage ranking dataset, which is provided by the authors[3]. We fine-tuned their model on the WWW-2 test collection [9]. Documents were split into a maximum of 16 passages, using a sliding window of 225 tokens with a stride of 200 tokens to fit in the model.

Table 2: Evaluation results of our submitted runs and the baseline based on the Gold file. The best result among the NEW runs in a column is in bold.

Run	nDCG	Q	nERR	iRBU
SLWWW-CO-REP-1	0.3686	0.2886	0.5098	0.7840
SLWWW-CO-NEW-2	0.3398	0.2718	0.5129	0.7358
SLWWW-CO-NEW-3	0.3388	0.2670	**0.5248**	0.7368
SLWWW-CO-NEW-4	**0.3650**	**0.2891**	0.5052	**0.7986**
SLWWW-CO-NEW-5	0.3193	0.2538	0.4288	0.7133
baseline	0.3205	0.2473	0.4541	0.7327

Table 3: Evaluation results of our submitted runs and the baseline based on the Bronze-All file. The best result among the NEW runs in a column is in bold.

Run	nDCG	Q	nERR	iRBU
SLWWW-CO-REP-1	0.5846	0.5629	0.7537	0.9397
SLWWW-CO-NEW-2	0.5600	0.5316	**0.7330**	**0.9244**
SLWWW-CO-NEW-3	0.5464	0.5137	0.7242	0.9192
SLWWW-CO-NEW-4	**0.5750**	**0.5397**	0.7209	0.9213
SLWWW-CO-NEW-5	0.5410	0.5113	0.6939	0.8888
baseline	0.5170	0.4806	0.6711	0.8920

2.2.3 REP run. We used Birch [1] for implementation. We first retrieved the top 1,000 relevant documents for each topic as described in the above section. We then used the provided BERT-Large model[4] which is first fine-tuned on the MSMARCO dataset and later fine-tuned on the MB dataset [8]. We measured the performance on the Robust04 dataset [13] to determine the hyper-parameter settings as in KASYS work [12].

[2] http://lemurproject.org/clueweb12/

[3] https://github.com/canjiali/PARADE

[4] https://github.com/castorini/birch

Table 4: Evaluation results of our REP run and the KASYS's REV run based on the Gold file.

Run	nDCG	Q	nERR	iRBU
SLWWW-CO-REP-1	0.3686	0.2886	0.5098	0.7840
KASYS-CO-REV-6	0.3682	0.2890	0.5098	0.7811

Table 5: Evaluation results of our REP run and the KASYS's REV run based on the Bronze-All file.

Run	nDCG	Q	nERR	iRBU
SLWWW-CO-REP-1	0.5846	0.5629	0.7537	0.9397
KASYS-CO-REV-6	0.5931	0.5743	0.7634	0.9424

3 RESULTS

3.1 Evaluation results

Table 2 shows the results of our submitted runs and the baseline based on the Gold file. Table 3 shows the results of our submitted runs and the baseline based on the Bronze-All file. Baseline is an Anserini-based vanilla BM25 run provided by the task organizers. According to the overview paper [10], no statistically significant differences were observed between our runs. However, we can see that among the NEW runs, SLWWW-CO-NEW-4 performed the best in both results in terms of nDCG and Q. Specifically, we can observe that the iRBU of this run outperforms other NEW runs and the baseline by a relatively large amount. This shows that splitting documents into chunks did not contribute to the improvement of retrieval effectiveness. We can also see that using a larger subset corpus does not affect the search effectiveness as can be observed from the slight difference between SLWWW-CO-NEW-2 and SLWWW-CO-NEW-3. Furthermore, we can observe that our NEW runs based on COIL outperform the baseline, indicating the effectiveness of introducing contextualized representations into a lexical exact match framework.

Gold file-based evaluation results and Bronze-ALL file-based evaluation results of our REP run and KASYS team's REV run are shown in Table 4 and Table 5, respectively. We can see that the two runs performed very similarly, suggesting the success of reproduction to some degree.

3.2 Topic Analysis

We conducted a per-topic analysis for further discussion. The analysis is based on nDCG, and the Bronze-All file was used as the relevance assessment. To prevent redundancy, from this point forward, SLWWW-CO-REP-1 is abbreviated as Run 1 and SLWWW-CO-NEW-2 as Run 2, and so on.

3.2.1 Poorly performing topics overall. Table 6 shows the worst performing topics and the mean nDCG over all submitted runs. Of the five shown, “idf inventor" had nDCG of 0 for all runs. Analysis of this shows that documents that were scored highly by the models often had a different word abbreviation of “idf" rather than the inverse document frequency, which was the intent of the topic creator. For example, documents containing “India Design Forum", “Intel Developer Forum", “Israeli Defense Forces", etc. were scored highly. In addition, “idf" for inverse document frequency often appeared in the documents as “tf-idf", which may have been judged by the models to be different from the query intent. Only 6 of the 231 documents were assessed as relevant. Another poorly performing topic was “half life", where the topic creators intended to know about the story and characters of the game “Half-Life". The highly scored documents by the models included documents about radioactive half-life, the download page of the game, etc. These topics share the common characteristics of being short, ambiguous, and having multiple intentions. It would have been difficult for the models to capture the query creator's intent without using the description field.

Table 6: Mean nDCG of the poorly performing topics over all WWW-4 runs.

Topic	Content	nDCG
203	idf inventor	0.0000
220	half life	0.1491
234	Warriors v.s. NETS 2021	0.1517
228	block chain crypto	0.2392
206	DC and Marvel characters	0.2814

Table 7: Average document length and the number of topics when Run 2 outperforms Run 4 and when Run 4 outperforms Run 2.

Better run	Number of topics	Average document length
Run 2	20	3504
Run 4	28	2731
All topics	50	2998

Table 8: Average document length and the number of topics when Run 4 outperforms Run 5 and when Run 5 outperforms Run 4.

Better run	Number of topics	Average document length
Run 4	27	2666
Run 5	22	3489
All topics	50	2998

3.2.2 Effect of document length. Run 2 and Run 5 are systems that have been devised to deal with the document length limitation of the language model, but Run 4 is not. Therefore, a comparison of these models was conducted to examine the impact of document length. We calculated the average document length of the top 1000 candidate documents for each topic extracted by BM25 for each topic, and averaged them on the topics where one Run outperform the other Run. Comparisons of Run 4 (The original COIL system) and Run 2 (COIL system with document splitting), and Run 4 and Run 5 (PARADE system) are shown in Table 7 and Table 8, respectively. We can see that the document lengths for topics for which

Table 9: nDCG of the topics where Run 4 significantly outperformed the baseline.

Topic	Content	Run 4	baseline	difference
217	inventor of the Web	0.7461	0.0568	0.6893
234	Warriors v.s. NETS 2021	0.5273	0.0000	0.5273

Table 10: nDCG of the topics where the baseline significantly outperformed Run 4.

Topic	Content	Run 4	baseline	difference
210	hypothermia treatment	0.1357	0.6303	0.4946
240	what is clickbait	0.3748	0.8249	0.4501

Run 2 and Run 5 perform better than Run 4 are longer than those for which Run 4 performs better. This indicates that Run 2 and Run 5 were able to handle long documents well. However, the number of topics that Run 4 outperforms the other run is more than that of the other way round. This might imply that there are many cases where the first 512 tokens of a document are sufficient to determine its compatibility.

3.2.3 Comparison of COIL and BM25. COIL is a system that introduced contextualized representation into lexical match retrievers such as BM25. Table 9 shows the topics and nDCG where Run 4 (COIL) significantly outperformed the baseline (BM25). The topic with the largest difference between the two runs was “inventor of the web". The highly scored documents by the baseline included the term “inventor" and “web" a lot, but not the key terms such as “World Wide Web" or “WWW". For this topic, it can be said that COIL succeeded in capturing the topic creator’s intent, which was not captured by BM25, because of the contextual information.

Table 10 shows the topics and nDCG where the baseline significantly outperformed Run 4. The topic with the largest difference where the baseline outperformed Run 4 was “hypothermia treatment". The highly scored documents by Run 4 included many documents about therapeutic hypothermia, however, the intent of this topic was to know how to deal with hypothermia when it occurs. It is a topic that could be taken either way and perhaps the consideration of contextual information led to hypothermia therapy, which was unfortunate in this case.

4 CONCLUSIONS

The SLWWW team participated in the NTCIR-16 We Want Web with CENTRE (WWW-4) task. We submitted three NEW runs based on contextualized exact lexical match, and a NEW run based on the passage representation aggregation method. We also submitted a run aiming to reproduce the top-performing run in the last WWW task. With the per-topic analysis, we examined and discussed topics with poor overall results, the impact of document length, and the relationship between COIL and BM25. One of the future tasks includes creating a system that also uses the description field so that it can deal with topics that were poorly done in this task.

REFERENCES

[1] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval. 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)*. Association for Computational Linguistics, Hong Kong, China, 3490–3496. https://doi.org/10.18653/v1/D19-1352

[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. arXiv:1611.09268 [cs.CL]

[3] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. arXiv:2003.10555 [cs.CL]

[4] 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

[5] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List. arXiv:2104.07186 [cs.IR]

[6] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2021. PARADE: Passage Representation Aggregation for Document Reranking. arXiv:2008.09093 [cs.IR]

[7] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. arXiv:2010.06467 [cs.IR]

[8] Jimmy Lin, Yulu Wang, Miles Efron, and Garrick Sherman. 2014. Overview of the TREC-2014 Microblog Track. In *Text Retrieval Conference (TREC)*. https://trec.nist.gov/pubs/trec23/papers/overview-microblog.pdf

[9] Jiaxin Mao, Tetsuya Sakai, Cheng Luo, Peng Xiao, Yiqun Liu, and Zhicheng Dou. 2019. Overview of the NTCIR-14 We Want Web Task. In *14th NTCIR Conference*.

[10] Tetsuya Sakai, Sijie Tao, Zhumin Chu, Maria Maistro, Yujing Li, Nuo Chen, Nicola Ferro, Junjie Wang, Ian Soboroff, and Yiqun Liu. 2022. Overview of the NTCIR-16 We Want Web with CENTRE (WWW-4) Task. In *Proceedings of NTCIR-16*. to appear.

[11] Tetsuya Sakai, Sijie Tao, Zhaohao Zeng, Yukun Zheng, Jiaxin Mao, Zhumin Chu, Yiqun Liu, Maria Maistro, Zhicheng Dou, Nicola Ferro, and Ian Soboroff. 2020. Overview of the NTCIR-15 We Want Web with CENTRE (WWW-3) Task. In *Proceedings of NTCIR-15*. 219–234.

[12] Kohei Shinden, Atsuki Maruta, and Makoto P. Kato. 2020. KASYS at the NTCIR-15 WWW-3 Task. In *Proceedings of NTCIR-15*. 235–238.

[13] Ellen M. Voorhees. 2005. The TREC Robust Retrieval Track. *SIGIR Forum* 39, 1 (jun 2005), 11–20. https://doi.org/10.1145/1067268.1067272

[14] 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) *(SIGIR ’17)*. Association for Computing Machinery, New York, NY, USA, 1253–1256. https://doi.org/10.1145/3077136.3080721