NTCIR 17 Conference: Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, December 12-15, 2023, Tokyo, Japan
DOI: https://doi.org/10.20736/0002001331

KASYS at the NTCIR-17 Transfer Task

Kenya Abe
University of Tsukuba
s2321672@u.tsukuba.ac.jp

Kota Usuha
University of Tsukuba
s2221635@s.tsukuba.ac.jp

Makoto P. Kato
University of Tsukuba
mpkato@acm.org

ABSTRACT

This paper describes the KASYS team's participation in the NTCIR-17 Transfer Task. To generate our runs, we used neural IR models such as Contriever, ColBERT, and SPLADE with different fine-tuning strategies. One of the main purposes to participate in this task is to examine the generalizability of the neural IR models to Japanese document retrieval.

KEYWORDS

neural IR models, dense retriever, learned sparse retriever

TEAM NAME

KASYS

SUBTASKS

Dense First Stage Retrieval
Dense Reranking

1 INTRODUCTION

This paper describes the KASYS team's participation in the NTCIR-17 Transfer Task. We submitted nine runs for Dense First Stage Retrieval subtask and five runs for Dense Reranking subtask. One of the main purposes to participate in this task is to examine the potential of the first stage retriever for Japanese document retrieval.

2 METHODS

This section reviews neural IR models used for our runs.

2.1 Contriever

We submitted five runs based on Contriever [2]. Contriever is a dense retrieval model that has a bi-encoder architecture, employing the same encoder for both query encoder and document encoder. The model is designed with the objective of improving zero-shot retrieval effectiveness, pre-training with inverse Cloze task (ICT) [3] and contrastive learning using Momentum Contrast (MoCo) [4] as a contrastive learning framework.

ICT is a data augmentation technique constructing training samples by exclusively sampling two segments from a document. In the pre-training phase, this approach engages in contrastive learning with self-supervision, treating one segment as the query and the other as the positive example for learning embedding representations. Besides ICT, additional data augmentation techniques are applied, including generating queries and contexts through cropping twice from the text independently and the elimination of 10% of the tokens, to extend the dataset.

Contrastive learning is a methodology distinguishing between embedding representations of documents relevant to a query and those that are not. The contrastive loss L is computed using a query q, a document $d+$ relevant to q, and a set of K non-relevant document $d_1^-, d_2^-, \ldots, d_K^-$, as defined by the following equation:

$$L = -\frac{\exp(f(q, d^+)/\tau)}{\exp(f(q, d^+)/\tau) + \sum_{i=1}^{K} \exp(f(q, d_i^-)/\tau)}$$

The number of negative examples is important in contrastive learning, and various techniques have been proposed to manage large sets of negative examples. Within Contriever, MoCo is utilized as the framework for contrastive learning. MoCo stores the embedding representations of positive examples in a queue and uses them as negative examples while pre-training. To maintain consistency in the embedding representations of the negative examples, a Momentum Encoder is employed as the encoder for positive examples. The parameters of the Momentum Encoder are updated through a moving average with the parameters of the query encoder, ensuring consistency in the embedding representations of the negative examples.

Performance evaluation results have confirmed that Contriever outperforms existing dense retrieval models in terms of zero-shot retrieval performance. On average, it also surpasses the BM25. In multilingual settings, training data in 29 languages are generated from CCNet, with pre-training being conducted accordingly and outperformed BM25 with fine-tuning on MS MARCO.

2.2 ColBERT

We also submitted a single run based on ColBERT, as another dense retrieval model. ColBERT has a late-interaction architecture which computes the similarity between encoded representations of query and document tokens. Due to much computational cost, ColBERT employs a multi-stage retrieval architecture. At the first stage, candidate documents are retrieved by approximated nearest neighbor search. At the second stage, the candidate documents are reranked by the sum of maximum similarity between query and document tokens. Equation 1 is the score function to rerank the documents based on the similarity between query and document tokens. Let q represent a query and d represent a document. The query q is split into a sequence of tokens $q = \{q_1, q_2, ...q_{|q|}\}$ by a tokenizer. The document d is also split into a sequence of tokens $d = \{d_1, d_2, ...d_{|d|}\}$. The tokens are encoded into query and document embedding $\eta(q_i)$ and $\eta(d_j)$ by an encoder η (e.g., BERT). The score of d in response to q is defined as:

$$S_{q,d} = \sum_{i=1}^{|q|} \max_{j=1,...|j|} \eta(q_i)^T \eta(d_i)$$

This function computes the similarity of every query-document token pairs, getting the maximum similarity for every query token. The sum of the maximum similarity scores is used to rank the documents.

At training, ColBERT uses pairwise softmax cross-entropy as a loss function and MS MARCO triples[1] as training data. Triples consist of a query, a positive document and negative document. But, for non-English dense retrieval, we have no sufficient data for fine-tuning retrieval models. Therefore, we adopted the ideas of ColBERT-X [6], which is multilingual version of ColBERT [5] designed for Cross Language Information Retrieval(CLIR). In CLIR, given a query in language l, the system retrieves relevant documents written in language l'. The authors proposed a method called Zero-Shot and Translate-Train in this paper, aimed at enhancing the performance of the system in CLIR scenarios. Transfer Task focuses on searching for Japanese documents using Japanese queries. However, the subsequent discussion is conducted under the premise that English queries are employed to retrieve Japanese documents.

The former approach (Zero-Shot) fine-tunes an encoder such as XLM-RoBERTa and mBERT using English training data. This approach proves to be somewhat effective without training data in target language, because in pre-training phase, the pre-training data contrains languages other than English. Zero-Shot is trained to effectively search for English documents with English queries using MSMARCO, it is assumed to operate in a mono-lingual scenario during retrieval. Therefore, when we inference for Japanese documents with English queries, there would be a need to translate the English queries into Japanese at that time. Since Transfer Task is a mono-lingual scenario, we don't need to translate queries during the search process to apply Zero-Shot to this task.

The latter approach (Translate-Train) translates the English training data into the target language l (the language used in the collection) and uses it to train the encoder. In this approach, translated MS MARCO data is used. However, since the problem setting involves searching Japanese documents with English queries, only the documents are translated, and the queries remain in original (English) data for training. If all English data is translated to create a complete set of Japanese translated data, it enables us to perform the same processing as when training and searching with actual Japanese data.

2.3 SPLADE

We introduced SPLADE [7] [8], which is a state-of-the-art sparse lexical-based retriever. Unlike Contriever and ColBERT, SPLADE generates sparse representations by connecting a Masked Language Model (MLM) head to the last hidden layer of the encoder model. Let q represent a query and d represent a document. The tokens t (query or document) are split into a sequence of tokens $t = \{t_1, t_2, ...t_{|t|}\}$ by a tokenizer and encoded into query and document embedding $\{h_1, h_2, ...h_\}$. We calculate $w_{i,j}$the importance of vocabulary token j for sequence token i:

$$w_{i,j} = \phi(h_i)^T E_j + b_j \quad j \in \{1, ...|V|\}$$

After getting distribution of vocabulary token importance for each token in the sequence (importance matrix), we apply max pooling to vocabulary dimensions of importance matrix to compute token importance of $|V|$-dimensional sparse vector.

$$w_j = \max_i \log(1 + ReLU(w_{i,j}))$$

[1]https://microsoft.github.io/msmarco/Datasets.html

SPLADE uses the contrastive loss L, which is computed using a query q, a document $d+$ relevant to q, one hard negative d_1^-, and a set of K non-relevant document $d_2^-, d_3^-, \ldots, d_K^-$, as defined by the following equation:

$$L = -\frac{\exp(f(q, d^+)}{\exp(f(q, d^+)) + \sum_{i=1}^{K} \exp(f(q, d_i^-))}$$

To make a set of K non-relevant document, SPLADE employs In-batch negative sampling, which utilizes K-1 relevant documents from within the same batch of size K as negative documents.

We utilized the Japanese version of BERT to accommodate Japanese search queries and conducted training using data translated from MS MARCO into Japanese.

2.4 Reciprocal Rank Fusion

To validate the hybrid search in Japanese document retrieval, we submitted two ensemble methods based on reciprocal rank fusion. The first one (KASYS-First-7) is ColBERT-X + BM25, and the second one (KASYS-First-9) is ColBERT-X + SPLADE. Reciprocal rank fusion computes the score of each document as follows:

$$score(d) = \sum_{r \in R} \frac{1}{k + r(d)}$$

This function calculates the score of document d based on its rank $r(d)$ within multiple ranking lists R. Here, k is a parameter used for weighting the ranks, with a smaller k value emphasizing higher ranks. For each ranking list r the rank $r(d)$ of the document d is added to k and then inverted. The sum of these inverted values across all ranking lists gives the final score of document d.

3 RUNS

3.1 Dense First Stage Retrieval Subtask

We describe details of each run submitted to the Dense First Stage Retrieval subtask.

3.1.1 KASYS-First-1. We used Contriever without fine-tuning. Pre-training was performed on Wikipedia and CCNet.

3.1.2 KASYS-First-2. We used Contriever fine-tuned on MS MARCO.

3.1.3 KASYS-First-3, -4, and -5. We used Contriever fine-tuned on MS MARCO and NTCIR-1 dataset. These three runs differ in the way that they sampled negative examples used for fine-tuning on NTCIR-1. When fine-tuning the model on NTCIR-1, 53 queries are used for training and 30 queries for validation.

For KASYS-First-3, documents in the same batch aside of the gold document are used as negative samples. For KASYS-First-4, documents judged as irrelevant for the query is used as negative samples. For KASYS-First-5, we firstly mined negative documents using Contriever fine-tuned on MS MARCO and used them as negative samples.

3.1.4 KASYS-First-6 and -8. For KASYS-First-6, we use ColBERT-X trained with both MS MARCO and mmarco dataset. This method can be considered as an intermediate approach between Zero-Shot and Translate-Train. For KASYS-First-8, we use SPLADE trained with mmarco dataset and cl-tohoku/bert-base-japanese-v3 as an encoder.

Run name	Description
KASYS-First-1	Contriever (no fine-tuning)
KASYS-First-2	Contriever (fine-tuning on MS MARCO)
KASYS-First-3	Contriever (fine-tuned on MS MARCO and NTCIR-1 with random negatives)
KASYS-First-4	Contriever (fine-tuned on MS MARCO and NTCIR-1 with qrel negatives)
KASYS-First-5	Contriever (fine-tuned on MS MARCO and NTCIR-1 with hard negatives)
KASYS-First-6	ColBERT-X (fine-tuned on MS MARCO and mmarco)
KASYS-First-7	ColBERT-X + BM25
KASYS-First-8	SPLADE (fine-tuned on mmarco)
KASYS-First-9	ColBERT-X + SPLADE
KASYS-Second-1	Reranking the BM25 baseline by KASYS-First-1
KASYS-Second-2	Reranking the BM25 baseline by KASYS-First-2
KASYS-Second-3	Reranking the BM25 baseline by KASYS-First-3
KASYS-Second-4	Reranking the BM25 baseline by KASYS-First-4
KASYS-Second-5	Reranking the BM25 baseline by KASYS-First-5

Table 1: KASYS runs.

3.1.5 KASYS-First-7 and -9. For KASYS-First-7, we implemented a hybrid approach [2] combining BM25 and ColBERT-X. For KASYS-First-9, we employed another hybrid method that integrates SPLADE and ColBERT-X. In both cases, we utilized the runs submitted for ColBERT-X and SPLADE in this study. As for BM25, we relied on the search results produced using the method provided by the organizers.

3.2 Dense Reranking Subtask

In the Reranking Subtask, the BM25 baseline is reranked using KASYS-First-1, -2, -3, -4, and -5, resulting in the generation of runs designated as KASYS-Second-1, -2, -3, -4, and -5, respectively.

4 EXPERIMENTS

The results of the Dense First Stage Retrieval Subtask are presented in Table 2. KASYS-First-7, which utilized RRF between BM25 and ColBERT-X, exhibited the highest performance. KASYS-First-9, which applied RRF with SPLADE and ColBERT-X, underperformed KASYS-First-6, a method solely based on ColBERT-X, except for nDCG@15 and nDCG@20. Among the methods using Contriever, KASYS-First-5, a method that fine-tuned with hard negatives, showed the highest performance, except for nDCG@15. In the Dense Reranking Subtask, KASYS-Second-5, which was fine-tuned with hard negatives, demonstrated the highest performance, excluding RR and nDCG@5 (Table 3).

5 CONCLUSIONS

In this paper, we explored several techniques for information retrieval in Japanese. Specifically, we experimented with various fine-tuning approaches for the Contriever, while separately evaluating other retrieval models like ColBERT-X and SPLADE. We also investigated hybrid approaches combining these models. Our experiments demonstrated that even with limited data, fine-tuning on test collection can significantly improve retrieval results. Notably, hard negative sampling, which is generally considered effective, proved to be the most beneficial in our context. Regarding the performance of individual retrieval models, our results suggest that the late-interaction model, ColBERT-X, might be the most effective. Despite not being fine-tuned with the test collection, ColBERT-X achieved the highest score among standalone techniques. We expect that its performance can be further boosted with appropriate fine-tuning. In our examination of hybrid methods, the combination of ColBERT-X and BM25 yielded the highest score. This result aligns with the general understanding that integrating dense and sparse retrieval models can produce more refined results.

These insights lead us to think that best practices and findings, which are considered effective in general, are also likely to work well for Japanese information retrieval. For our future work, we plan to implement fine-tuning for ColBERT-X, and we are also considering the adoption of other retrieval models that were not explored in this study.

[2]For the Reciprocal Rank Fusion (RRF), we set k to 20.

ACKNOWLEDGMENTS

We thank the NTCIR-17 Transfer task organizers for organizing the task and preparing test collections.

REFERENCES

[1] Joho, H., Keyaki, A., and Oba, Y. 2023. Overview of the NTCIR-17 Transfer Task. In Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, December 12-15, 2023, Tokyo, Japan. https://doi.org/10.20736/0002001319

[2] Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. (2021). Unsupervised dense information retrieval with contrastive learning. arXiv preprint arXiv:2112.09118.

[3] Lee, K., Chang, M. W., and Toutanova, K. (2019). Latent retrieval for weakly supervised open domain question answering. arXiv preprint arXiv:1906.00300.

[4] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R.B. (2019). Momentum Contrast for Unsupervised Visual Representation Learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9726-9735.

[5] Khattab, Omar, and Matei Zaharia. "Colbert: Efficient and effective passage search via contextualized late interaction over bert." Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 2020.

[6] Nair, Suraj, et al. "Transfer learning approaches for building cross-language dense retrieval models." European Conference on Information Retrieval. Cham: Springer International Publishing, 2022.

Run name	nDCG@5	nDCG@10	nDCG@15	nDCG@20	nDCG@1000	RR	MAP
KASYS-First-1	0.360	0.318	0.289	0.275	0.296	0.530	0.088
KASYS-First-2	0.529	0.446	0.407	0.372	0.358	0.684	0.128
KASYS-First-3	0.538	0.453	0.404	0.376	0.362	0.681	0.131
KASYS-First-4	0.538	0.450	0.403	0.376	0.362	0.680	0.131
KASYS-First-5	0.540	0.462	0.406	0.383	0.367	0.688	0.135
KASYS-First-6	0.599	0.539	0.497	0.468		0.728	0.229
KASYS-First-7	**0.601**	**0.562**	**0.518**	**0.502**		**0.735**	**0.263**
KASYS-First-8	0.488	0.452	0.423	0.404		0.677	0.157
KASYS-First-9	0.563	0.526	0.506	0.482		0.681	0.227

Table 2: Retrieval effectiveness of our runs in Dense First Stage Retrieval Subtask.

Run name	nDCG@5	nDCG@10	nDCG@15	nDCG@20	nDCG@1000	map	RR
KASYS-Second-1	0.450	0.384	0.356	0.341	0.243	0.105	0.588
KASYS-Second-2	0.559	0.471	0.431	0.405	0.280	0.133	0.716
KASYS-Second-3	**0.586**	0.481	0.438	0.418	0.285	0.136	**0.745**
KASYS-Second-4	0.579	0.477	0.434	0.415	0.283	0.135	0.727
KASYS-Second-5	0.575	**0.487**	**0.441**	**0.419**	**0.289**	**0.138**	0.741

Table 3: Retrieval effectiveness of our runs in Dense Reranking Subtask.

[7] Formal, Thibault, Benjamin Piwowarski, and Stéphane Clinchant. "SPLADE: Sparse lexical and expansion model for first stage ranking." Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021.

[8] Formal, Thibault, et al. "SPLADE v2: Sparse lexical and expansion model for information retrieval." arXiv preprint arXiv:2109.10086 (2021).