JaColBERT and Hard Negatives, Towards Better Japanese-First Embeddings for Retrieval: Early Technical Report

Benjamin Clavié

ben@clavie.eu

Abstract. Document retrieval in many languages has been largely relying on multi-lingual models, and leveraging the vast wealth of English training data. In Japanese, the best performing deep-learning based retrieval approaches rely on multilingual dense embeddings. In this work, we introduce (1) a hard-negative augmented version of the Japanese MMARCO dataset and (2) JaColBERT, a document retrieval model built on the ColBERT model architecture, specifically for Japanese. JaColBERT vastly outperform all previous monolingual retrieval approaches and competes with the best multilingual methods, despite unfavourable evaluation settings (out-of-domain vs. in-domain for the multilingual models). JaColBERT reaches an average Recall@10 of 0.813, noticeably ahead of the previous monolingual best-performing model (0.716) and only slightly behind multilingual-e5-base (0.820). These results are achieved using only a limited, entirely Japanese, training set, more than two orders of magnitudes smaller than multilingual embedding models. We believe these results show great promise to support retrieval-enhanced application pipelines in a wide variety of domains.

Keywords: Japanese NLP · Dense Retrieval · Retrieval-Augmented Generation (RAG) · ColBERT · Sentence Embeddings

1 Contributions

In this first version of this work, we:

- Release a dataset of hard-negatives for the Japanese language subset of MMArco[1]
- Release JaColBERT[2], a Japanese-only version of ColBERT [6] trained on the dataset above and outperforming all existing Japanese models on retrieval tasks and competitive with multilingual e5 models, despite the latter having been trained on the training sets associated with our evaluation data.

[1] https://huggingface.co/datasets/bclavie/mmarco-japanese-hard-negatives
[2] https://huggingface.co/bclavie/JaColBERT

2 Data

There is a growing number of Japanese NLP datasets [8,2,14], a lot of them introduced through constrained automatic translation methods in order to leverage the vast wealth of data annotated for English.

However, as of yet, there appears to have been a lack of large scale datasets to train generalist retrieval models. The release of MMARCO and its japanese subsplit [1], the multi-lingual version of the MS MARCO Document Retrieval dataset [11], has provided an initial large scale dataset to be used for this purpose.

MS MARCO is one of the most widely used dataset for training document embedding models, and has been shown to provide models with impressive generalisation on a wide variety of retrieval tasks [6,12,16,15], such as the ones in the BEIR benchmark [13].

2.1 Generating Japanese Hard Negatives

In both the existing literature [18] and informal discussions, the importance of hard negatives in training retrieval models is highlighted as particularly important. A *hard negative* is a negative example that looks very similar to a positive example, and serves to improve a model's ability to discriminate between relevant and "relevant-looking" irrelevant passages.

There are many ways of generating hard negatives. Human annotation, while excellent, is prohibitively time-consuming and costly at the scale required, thus, hard negatives are generally generated by existing retrieval methods, both sparse (BM25...) and dense, such as document embedding models or cross-encoders.

To support the development of stronger Japanese retrieval models, we generate hard negatives for the MMARCO dataset, using two approaches:

Multilingual e5 embeddings The current leading multilingual dense document embeddings, with a strong variety on many languages, including Japanese. We embed the entirety of the MMARCO Japanese Corpus, then retrieve the 110 most similar documents for each of them. We discard the 10 most similar documents, as MMARCO is a lossy dataset: some passages for a query are not annotated as positive examples, although they would indeed be considered relevant. Discarding the most similar documents help us avoid integrating these false negatives to our training data. Finally, we randomly sample 25 examples, and choose them as our e5-generated hard negatives.

BM25 We use the Anserini [10] implementation of BM25, as well as their default Japanese Analyser. For each individual query, we retrieve up to 10 similar documents, once again discarding the first ten matches.

The generated data is used to train JaColBERT, along with the initial training negatives provided in the original dataset. We make our full dataset available to support future work [3].

[3] https://huggingface.co/datasets/bclavie/mmarco-japanese-hard-negatives

Recently, the release of MIRACL [20] and Mr.TyDi [19], two multilingual information retrieval datasets, have also provided us with large corpora and a small subset of annotated positive examples. In future work, we believe it would be useful to generate hard-negatives for those datasets on a large scale, to further diversify training sets. We do not do so in this work due to both limited compute and wanting to keep both of those datasets unseen for out-of-domain evaluation of JaColBERT. We also do not explore the Japanese AIO QA retrieval competition datasets, which could provide another useful data source in the future.

3 Embeddings & Retrieval

Document retrieval, particularly in the context of RAG (Retrieval-Augmented Generation) pipelines, has emerged as an increasingly important topic at the intersection of NLP and Information Retrieval.

Most retrieval methods have strong tradeoffs:

- Traditional sparse approaches, such as BM25, are strong baselines, **but** do not leverage any semantic understanding, and thus hit a hard ceiling.
- Cross-encoder retriever methods are powerful, **but** prohibitively expensive over large datasets: they must process the query against every single known document to be able to output scores.
- Dense retrieval methods, using dense embeddings in vector databases, are lightweight and perform well, **but** are data-inefficient (they require hundreds of millions if not billions of training examples pairs to reach state-of-the-art performance) and generalise poorly in a lot of cases, as representing every single aspect of a document, to be able to match it to any related query, into a single vector is not a solved problem.

Recent work has focused on attempting to leverage the benefits of both sparse and dense retrieval methods. This very recent line of work has produced very capable models such as SPLADE [4], ColBERT [6] and SparseEmbed [7].

Specifically, ColBERT, and its second version [12], leverage multiple tricks to build upon the strong representation power of transformer models such as BERT [3] to represent documents as bags-of-centroids by representing documents as being composed of many smaller contextualised vectors, rather than a single, large dense representation.

In this work, we introduce JaColBERT, building upon ColBERT to efficiently train a Japanese retrieval model, without the need of billion training examples from datasets in other languages.

3.1 JaColBERT

Leveraging the data and model architectures described above, we introduce **JaColBERT**. Our approach builds upon ColBERT to efficiently train a Japanese retrieval model, without the need of billion training examples from multilingual training sets.

To train this initial version of JaColBERT, we randomly sample ten million *(Query, PositivePassage, NegativePassage)* triplets from our hard-negative augmented MMARCO dataset to serve as our training step. Training on those ten million triplets takes 10 hours.

We initialise JaColBERT from Tohoku University's BERT-BASE-JAPANESE-V3. BERT is generally considered as the strongest base model for ColBERT models, as RoBERTa has been anecdotally noted to struggle to learn the kind of representation needed for this approach to work. As a result, we do not evaluate Waseda University's Japanese RoBERTa. We have not yet evaluated Studio Ousia's LUKE [17] models either, although they may be a suitable abse model for this approach.

We train JaColBERT on 8 NVidia L4 GPUs, with a total batch size of 128 (16 per GPU). We perform training for the full number of steps to iterate over all the training pairs once (roughly 78 000). The model is trained with 8000 warm-up steps, and a learning rate of 5e-6. Our experiments showed worse performance with other common learning rates, with 3e-6 being the closest. Learning rates of 1e-5 and 2e-5 resulted in noticeable performance degradation at early evaluation steps, and were not evaluated further.

We set the maximum query length to 64, and the maximum document length to 228. As per ColBERTv2 [12], each vector representation, when indexing documents, is compressed to 2-bits, allowing for efficient storage of large volumes of data, with no impact on retrieval performance.

3.2 Fio Embeddings

As an early precursor to this work, we also released FIO-BASE-JAPANESE-V0.1 (Fio). Fio is initiated from BERT-BASE-JAPANESE-V3, trained for three epochs on JNLI [8] and JSNLI [14], then fine-tuned for a single epoch (due to compute constraints) on an extremely small subset (100 000 sentence pairs, half negatives and half positives) of MMARCO, as well as subsamples of MIRACL and Mr.TiDy. Fio is trained using AnglE optimisation [9]. More detail on Fio is outside the scope of this report and available on the associated release blog post.

At this stage, Fio for retrieval remains a proof of concept and should not be used in lieu of JaColBERT or multilingual e5 models on these tasks. However, we believe that the data released with this work should allow to easily train a version a monolingual dense embedding model with strong retrieval performance, and intend to do so in the future.

4 Evaluation

We evaluate various models, including JaColBERT, on three datasets: two document retrieval ones (MIRACL and Mr.TyDi) and a Question-Answering one

(JSQuAD). We report the recall@K for k={3, 5, 10} for the retrieval datsets and recall@K for k={1, 5, 10} for JSQuAD. We do not report Recall@1 on MIRACL and as it can be a flawed metric, since retrieval datasets are known for the presence of false negatives. JSQuAD, on the other hand, asks questions relevant to a specific context and has a low volume corpus, so we choose to report recall@1.

Our evaluation approach is based on the one of Nouu.Me [4], modified to support additional retrieval methods and datasets. We make the exact version of our evaluation code available [5].

4.1 Datasets and set-up

To speed up evaluation on limited hardware, we evaluate in the following setting:[6]
JSQuAD [8] We use the validation split, as the test split was not available at the time of this work. Passages explicitly listed as containing an answer for the query are treated as relevant passages, and every other passage is considered irrelevant. The total document count is 1145 documents.
MIRACL [20] We use the 860 evaluation queries provided. The associated positive passages are considered relevant. We use the validation split as hard negatives are readily available for it in benchmarks. For each query, we sample the top two hundred associated hard negatives [7]. Duplicates are removed. The total resulting document count is 156722.
Mr.TyDi (test set) [19] We use the provided 720 evaluation queries and their associated positive examples. We additionally sample a random hundred thousand passages from the full corpus, to serve as negative examples. The total document count is 100242.

	JSQuAD	MIRACL	Mr.TyDi
ColBERT-Japanese	Unseen	Unseen	Unseen
multilingual-e5-*	**Partially Seen (English version)**	**Seen**	**Seen**
sentence-bert-base-ja-*	?	Unseen	?
sup-simcse-ja-*	Unseen	Unseen	Unseen
fio-base-japanese-v0.1	Unseen	**Seen**	**Seen**
GLuCoSE-base-ja	Unseen	Unseen	**Seen**

Table 1. List of models we evaluate and whether or not a given task is in-domain (training set **was** used during training) or not (training set **was not** used during training).

[4] https://github.com/nouu-me/document_vector_search_benchmark
[5] https://github.com/bclavie/document_vector_search_benchmark
[6] Anytime random sampling is mentioned, it is initialised with the random seed 42.
[7] Hard negatives are provided by http://github.com/oshizo/JapaneseEmbeddingEval

4.2 Models

On top of our models, we evaluate an array of existing japanese document representation models: the best performing of Nagoya University's simcse-ja family of embeddings models (sup-simcse-ja-base and sup-simcse-ja-large) [14], GLuCoSE-base-ja[8] and sentence-bert-base-ja-*-v2 models[9]. We also report results for the current best-performing embedding models for Japanese Document Retrieval, the multilingual-e5 [15] family of models.
Unlike the other models, the evaluated e5 models are multilingual in nature, not Japanese-specific, and **have been previously exposed to all three of our evaluation datasets during training**. Table 1 provides an overview of which evaluation tasks the various models have been exposed to during training [10].

5 Results and Discussion

	JSQuAD			MIRACL			MrTyDi			Average		
	R@1	R@5	R@10	R@3	R@5	R@10	R@3	R@5	R@10	R@{1—3}	R@5	R@10
JaColBERT	**0.906**	**0.968**	**0.978**	0.464	0.546	0.645	0.744	0.781	0.821	0.705	0.765	0.813
m-e5-large (in-domain)	*0.865*	*0.966*	*0.977*	***0.522***	***0.600***	***0.697***	***0.813***	***0.856***	***0.893***	***0.73***	***0.807***	***0.856***
m-e5-base (in-domain)	*0.838*	*0.955*	*0.973*	*0.482*	*0.553*	*0.632*	*0.777*	*0.815*	*0.857*	*0.699*	*0.775*	*0.820*
m-e5-small (in-domain)	*0.840*	*0.954*	*0.973*	*0.464*	*0.540*	*0.640*	*0.767*	*0.794*	*0.844*	*0.690*	*0.763*	*0.819*
GLuCoSE	0.645	0.846	0.897	0.369	0.432	0.515	*0.617*	*0.670*	*0.735*	0.544	0.649	0.716
sentence-bert -base-ja-v2	0.654	0.863	0.914	0.172	0.224	0.338	0.488	0.549	0.611	0.435	0.545	0.621
sup-simcse -ja-base	0.632	0.849	0.897	0.133	0.177	0.264	0.454	0.514	0.580	0.406	0.513	0.580
sup-simcse -ja-large	0.603	0.833	0.889	0.159	0.212	0.295	0.457	0.517	0.581	0.406	0.521	0.588
fio-base-v0.1	0.700	0.879	0.924	*0.279*	*0.358*	*0.462*	*0.582*	*0.649*	*0.712*	*0.520*	*0.629*	*0.699*

Table 2. Results on three retrieval tasks for each model. In **bold** is the overall best result. Underlined is the best result for mono-lingual models. Results in *italic* indicate the model was exposed to the task evaluated during its training. Models denoted as (in-domain) were exposed to all three tasks during their training.

JaColBERT considerably outperforms all existing embedding approaches evaluated on all three tasks, despite all three of them being out-of-domain. Al-

[8] https://huggingface.co/pkshatech/GLuCoSE-base-ja
[9] https://huggingface.co/sonoisa/sentence-bert-base-ja-mean-tokens-v2
[10] Information is not provided for the sentence-bert-base-ja-* models, however they were trained prior the release of MIRACL.

though all three of these tasks are general domain, this suggests strong generalisation potential, with only light fine-tuning potentially needed. This makes JaColBERT a strong candidate for a variety of use-cases involving document retrieval, and easily integrable into RAG pipelines through its strong synergy with the DSPy (Demonstrate-Search-Predict) [5] approach.

On MIRACL and Mr.TyDi, JaColBERT's overall performance lags slightly behind both the base and small multi-lingual e5 models on MIRACL and Mr.TyDi. This small gap in performance can likely be at least partially explained by e5 models having been trained on the training set of both datasets.

The difference is greater with the large version of multilingual-e5, possibly due to the larger model being able to make better use of its large pretraining data and in-domain knowledge, although JaColBERT remains not far behind, and considerably closer than all previous Japanese-based approaches. Noticeably, on the dataset which e5 has had no direct exposure to, JSQuAD, JaColBERT outperforms multilingual-e5-large.

Moreover, the relatively small performance delta between our approach and these existing models is notable, as JaColBERT has been trained on just 10M triplets for 10 hours on 8 GPUs. On the other hand, multilingual e5 models are the result of an extensive two-step training process with unsupervised training on more than 3.8B sentence pairs followed by supervised training on a variety of retrieval and language entailment datasets.

This highlights the strong potential of ColBERT-based retrieval approaches, and greatly reduces the need to rely on extremely costly pre-training datasets to obtain satisfactory results. We believe that this model, as well as future iterations of it, are a strong first step towards supporting generalisable Japanese document retrieval with Japanese-only ressources and lower amounts of compute.

6 Final Word, Conclusion and Future Work

Thank you for reading this report. I hope it proves to be useful for your research or applications, and I'd highly encourage any researcher or practitioner wanting to further build upon this to reach out!

In this work, we have built upon the wealth of work existing in NLP, IR, and Japanese-specific NLP to produce both an improved Japanese retrieval training dataset, augmented with hard negatives, and the best monolingual Japanese document retrieval model on three benchmarks. This model, JaColBERT, leverages the ColBERT model architecture to create efficient Japanese document representation, optimised for retrieval. It significantly outperforms all existing Japanese embedding approaches, and comes closer to matching multilingual models trained on vastly larger amounts of data, despite being evaluated out-of-domain on benchmarks which are in-domain for the multilingual models.

We release both our augmented training dataset and JaColBERT, to support applications and future research.

Our work also highlights the many shortcomings of our current approach. Our data augmentation techniques, training methods as well as training time are constrained by limited resources, and further work could considerably improve performance.

Notably, this work only leverages 10 million training triplets from MMARCO, generates hard negatives in a naive way and does not use or augment any of the other common retrieval datasets. We also do not use the scoring generated from already strong models as teachers for JaColBERT, which has been shown to improve retrieval performance [12]. Finally, our evaluation relies on subsamples of large-scale datasets, and evaluating on full-size benchmarks could yield more insight. We plan on exploring this in future work.

References

1. Bonifacio, L., Jeronymo, V., Abonizio, H.Q., Campiotti, I., Fadaee, M., Lotufo, R., Nogueira, R.: MMARCO: A multilingual version of the MS MMARCO passage ranking dataset. arXiv preprint arXiv:2108.13897 (2021)
2. Chen, Z., Handa, H., Shirahama, K.: JCSE: Contrastive learning of japanese sentence embeddings and its applications (2023)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Formal, T., Lassance, C., Piwowarski, B., Clinchant, S.: SPLADE v2: Sparse lexical and expansion model for information retrieval. arXiv preprint arXiv:2109.10086 (2021)
5. Khattab, O., Singhvi, A., Maheshwari, P., Zhang, Z., Santhanam, K., Vardhamanan, S., Haq, S., Sharma, A., Joshi, T.T., Moazam, H., Miller, H., Zaharia, M., Potts, C.: DSPy: Compiling declarative language model calls into self-improving pipelines (2023)
6. Khattab, O., Zaharia, M.: Colbert: Efficient and effective passage search via contextualized late interaction over bert. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 39–48 (2020)
7. Kong, W., Dudek, J.M., Li, C., Zhang, M., Bendersky, M.: SparseEmbed: Learning sparse lexical representations with contextual embeddings for retrieval. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2399–2403 (2023)
8. Kurihara, K., Kawahara, D., Shibata, T.: JGLUE: Japanese general language understanding evaluation. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference. pp. 2957–2966 (2022)
9. Li, X., Li, J.: AnglE-optimized text embeddings (2023)
10. Lin, J., Crane, M., Trotman, A., Callan, J., Chattopadhyaya, I., Foley, J., Ingersoll, G., Macdonald, C., Vigna, S.: Toward reproducible baselines: The open-source ir reproducibility challenge. In: Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38. pp. 408–420. Springer (2016)
11. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: A human generated machine reading comprehension dataset. choice **2640**, 660 (2016)

12. Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., Zaharia, M.: ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 3715–3734 (2022)
13. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. arXiv preprint arXiv:2104.08663 (2021)
14. Tsukagoshi, H., Sasano, R., Takeda, K.: Japanese SimCSE technical report. arXiv preprint arXiv:2310.19349 (2023)
15. Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., Wei, F.: Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:2212.03533 (2022)
16. Xiao, S., Liu, Z., Zhang, P., Muennighof, N.: C-pack: Packaged resources to advance general chinese embedding. arXiv preprint arXiv:2309.07597 (2023)
17. Yamada, I., Asai, A., Shindo, H., Takeda, H., Matsumoto, Y.: LUKE: Deep contextualized entity representations with entity-aware self-attention. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6442–6454 (2020)
18. Zhan, J., Mao, J., Liu, Y., Guo, J., Zhang, M., Ma, S.: Optimizing dense retrieval model training with hard negatives. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1503–1512 (2021)
19. Zhang, X., Ma, X., Shi, P., Lin, J.: Mr. TyDi: A multi-lingual benchmark for dense retrieval. In: Proceedings of the 1st Workshop on Multilingual Representation Learning. pp. 127–137 (2021)
20. Zhang, X., Thakur, N., Ogundepo, O., Kamalloo, E., Alfonso-Hermelo, D., Li, X., Liu, Q., Rezagholizadeh, M., Lin, J.: MIRACL: A multilingual retrieval dataset covering 18 diverse languages. Transactions of the Association for Computational Linguistics **11**, 1114–1131 (2023)