IRLab-AMS at TREC'24 NeuCLIR Track

Jia-Huei Ju
University of Amsterdam, The Netherlands

Andrew Yates
Johns Hopkins University, United States

Abstract

In this notebook paper, we describe our participation as IRLab-AMS in the NeuCLIR. Our submitted results for two tasks, multi-lingual information retrieval (MLIR) and cross-language report generation (ReportGen). For MLIR, we explore the learned sparse representations with multi-lingual retrieval settings. For ReportGen, we experiment with several pipelines for generating long-form reports, including standard retrieval-augmented generation (RAG) and post-hoc citation methods. Additionally, we add an extra retrieval augmentation module to handle the limitation of ad-hoc retriever. The module can serve as distinct purposes, including relevance ranking, novelty ranking, and summarization, or by combining them.

CCS Concepts

• **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

ACM Reference Format:
Jia-Huei Ju and Andrew Yates. 2018. IRLab-AMS at TREC'24 NeuCLIR Track. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

TREC NeuCLIR track benchmarked diverse cross-language information retrieval applications over the years, including cross-lingual (CLIR) and multi-lingual (MLIR) document ranking. This year, the cross-language report generation (ReportGen) task is introduced as an extended application for CLIR. The task is designed to gather information in other languages and shrink language barriers by organizing retrieved documents into reports. We in this year submit results for both MLIR and ReportGen tasks. For MLIR, we experiment with learned sparse retrieval with multilingual language modeling. For ReportGen, we explore several potential pipelines for generating reports, including standard retrieval-augmented generation (RAG) and citation retrieval methods. Moreover, to better understand interactions between retrieval and generation, we equip the RAG pipeline with an extra augmentation module, aiming to bridge the information gap in between. And thus gain more insights from the empirical evaluations.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

2 Report Generation

In this section, we first describe the setups for report generation subtask in Section 2.1. Then, in Section 2.2, we introduce the retrieval context augmentation methods, and the fine-tuning strategies.

2.1 Preliminary

Let q denote a report request (in English) of users[1]. And $\mathcal{D}'$ refers to the document collection in targeted languages[2]. Based on the report request and the collection, the system first retrieves documents in retrieval context C, and then generate an attributed report r grounded on the source context, such as:

$$r = G(q, C), \quad C \leftarrow D = \mathit{Translate}\Big(D' \leftarrow F_\theta(q, \mathcal{D}'; k)\Big), \quad (1)$$

where F_θ is a cross-language retriever. While G is a generator and C is the texts derived from retrieved documents $d' \in D'$. $D' \subset \mathcal{D}'$ indicates the top-k retrieved documents. $\mathit{Translate}(\cdot)$ refers to an off-the-shelf translation API functions. That is, we transform the multilingual RAG task into a monolingual RAG task.

2.2 Retrieval Context Augmentation

On top of the standard *retrieve-translate-generate* pipeline (Eq. (1)), we explore the potential of incorporating additional retrieval augmentation modules to better understand the interactions between retrievers and generators.

2.2.1 Retrieval Augmentation. We equip the standard pipeline with the proposed augmentation module A_ϕ as follows,

$$C_{aug} = A_\phi(c \in C), \quad (2)$$

where the new context C_{aug} is generated by the augmentation module A_ϕ. Before prompting generator G, the augmentation module can compress the retrieved documents and incorporate more documents ($n > k$) in practice. All of the above can be done within a text-generation process. We focus exclusively on monolingual settings and use translated versions of the documents in context C, leaving the investigation of cross-language impacts to future work. Specifically, the augmentation module serves as a multi-document reasoner. We integrate several manipulations for different purposes, such as "compression," "relevance ranking," and "novelty ranking."

2.2.2 Parallel-encoded Encoders. Motivated by Fusion-in-Decoder (FiD) [4], we adopt the encoder-decoder architecture [13] as the backbone of the augmentation module A_ϕ. For efficiency, we independently encode top-n retrieved documents using the transformer encoder. Once the documents are encoded into dense representations, a transformer decoder can generate the augmented contexts in an autoregressive manner. We formulate the entire process as:

$$C_{aug} = A_\phi^{\text{dec}}(H_1, H_2, ..., H_k), \quad H_i = A_\phi^{\text{enc}}(q, d_i), \quad \forall d_i \in D_c \quad (3)$$

[1] A request contains the user's "problem statement" and "background". We only consider the former for retrieval, while background is also included for generation.
[2] In this track, the target languages include Russian, Chinese, and Persian.

where $H_i \in \mathbf{R}^{|d_i| \times h}$ denotes the h-dimensional tokens representations of document d_i, and C_{aug} is the augmented context decoded from all top-n documents. It is worth noting that this approach is conceptually related to context compression [2, 8] and long-context language modeling [16]. These methods differ primarily in the form of the expected generated outputs.

2.2.3 Learning to augment retrieval context. We fine-tune the augmentation module on the synthesized datasets derived from the multi-document summarization dataset, Multi-News [3]. Each example in the dataset contains multiple documents D, all considered relevant. Among these relevant documents, we classify them into two groups: informative documents D^+ and redundant documents[3] $D^\emptyset$. We also include negative documents D^- mined by BM25, to serve as learning signals for relevance. Hence, we recast the retrieval context augmentation task as a text generation task:

$$C_{aug} = A_\phi^{\text{dec}}\big(\text{shuffle}(d_1^+, d_2^+, ..., d_i^-, ..., d_j^\emptyset ..., d_k^+)\big),$$

where $d^+, d^-, d^\emptyset$ represent relevant, irrelevant, and redundant documents, respectively. During training, we randomly shuffle the document order to help the model learn three types of manipulations: "summarization", "relevance ranking", "novelty ranking". The supervised fine-tuning targets are defined as a sequence: $C_q^* = [c_1, c_2, ..., c_k]$ following the conditions:

$$c_i = \begin{cases} \texttt{[i]}\ s_i & \text{if } d_i \in D \setminus D^\emptyset \\ \texttt{[i] irrelevant.} & \text{if } d_i \in D^- \\ \texttt{[i] redundant.} & \text{if } d_i \in D^\emptyset \end{cases},$$

where `[i]` represents the input document index (i.e., the order) after shuffling. While s is the summary generated by `Llama-3.1-70B` [12] models. To indicate the document relevance and novelty, we set the output as text, `irrlevant` and `redundant`. For example, the source input: $[1]\{d^+\}[2]\{d^\emptyset\}[3]\{d^-\}$, will be paired with the target output: $[1]\{s_1\}$`[2]irrelevant.[3]redundant.` This approach allows various forms of augmentation to be learned within a single generative module.

2.3 Evaluation

2.3.1 pipelines. We evaluate three different pipelines: 1) Standard retrieval-augmented generation (**std**), 2) Post-hoc citation (**PostCite**) (i.e., generate-then-cite), and 3) Retrieval-context augmentation (**MdComp**). To ensure a fair comparison between pipelines, and to properly evaluate proposed augmentation module, we fixed retrieval, translation, and generation components without fine-tuning:

- Cross-language retrieval F_θ: we use official retrieval API provided by the NeuCLIR organizers.[4].
- Document translation: we use API from Google translate for translating all retrieved documents.
- Text generator G: we use both ChatGPT and Llama 3.1 8B and 70B models [12].

[3]Redundant documents are defined as those that, although relevant, contain less informative content.

[4]According to the guideline, the search engine used multi-vector dense retrieval approaches, PLAIDX [9, 15], as backend.

C	$Score_{arg}$	$Prec_{cite}$	$Recall_{nug}$
MdComp	17.9 / 18.5 / 16.6	70.3 / 73.7 / 75.9	17.1 / 15.7 / 13.5
MdComp (w/ $d_\emptyset$)	15.9 / 18.5 / 14.4	66.5 / 73.9 / 74.4	15.0 / 16.8 / 15.1
ReComp	29.2 / 22.7 / 27.7	78.1 / 71.9 / 77.7	13.3 / 15.2 / 15.8
PostCite-v	08.7 / 07.4 / 12.1	38.9 / 43.9 / 42.3	05.8 / 05.0 / 10.5
PostCite	18.8 / 12.6 / 18.1	45.2 / 48.0 / 54.6	10.8 / 08.5 / 10.2
std	33.7 / 26.5 / 36.0	79.5 / 88.9 / 86.1	18.2 / 21.6 / 20.9
std (`Llama-70B`)	56.6 / 47.2 / 46.4	85.2 / 87.1 / 92.7	19.7 / 25.5 / 18.1

Table 1: Evaluation of the different retrieval context augmentation, across three languages (fas/rus/zho). Except for the last row, the others are using `Llama-3.1-8B` as a generator.

We initialize our encoder-decoder models as Flan-T5-large[5] and fine-tune as mentioned in Section 2.2.3.

2.3.2 Augmentation setting. For retrieval-context augmentation, we first group the top-30 retrieved documents into 10 groups, each consisting of three document candidates: the i^{th}, the $(10+i)^{th}$ and the $(20+i)^{th}$ retrieved documents. This setting enables the module to use the top-relevant document to identify irrelevant and redundant documents at the latter positions. We compare our proposed MdComp with **ReComp** [14]; both use top-30 documents as retrieval context for the downstream generator. Afterwards, we post-process the augmented context C_{aug} by discarding irrelevant and redundant documents, keeping only the remaining text as the final augmented context for the generator. **Std** indicates the standard RAG with top-10 documents. The PostCite setting uses ChatGPT to first generate a structured report. Each sentence is then treated as a query to cite supporting documents afterward. We use the same cross-language retrieval API to retrieve the top-30 documents in other languages. Additionally, we employ a multilingual NLI model[6] to re-rank the top-30 documents, which is denoted as PostCite-V. Both post citation settings use only the top-2 documents.

2.3.3 Main results. Table 1 presents the empirical evaluation of retrieval context augmentation strategies across three language, using the official metrics: $Score_{arg}$, $Prec_{cite}$, and $Recall_{nug}$. Among the methods compared, ReComp consistently outperforms MdComp and PostCite variants across all metrics, showing a notable advantage in generating higher-quality arguments, better citation precision, and nugget recall. The inclusion of document scores in MdComp (w/ $d_\emptyset$) results in only minor improvements, suggesting that the method struggles to identify redundant content effectively, implying suboptimal learning. PostCite and PostCite-v perform weakest across the board, especially in $Prec_{cite}$ and $Recall_{nug}$, highlighting the limitations of generating without retrieval, implying NeuCLIR evaluation is retrieval-intensive. Lastly, we found the standard approach (std) still surpasses many augmented methods, indicating that augmentation modules are not yet effective. The final row, using Llama-70B, shows a substantial performance boost, achieving the best results across nearly all metrics and languages.

[5]https://huggingface.co/google/flan-t5-large

[6]`MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7`

3 Multilingual Retrieval

We submitted three MLIR runs using different combinations of the title and description fields. All three runs were automatic learned sparse retrieval runs using English queries and documents in their native languages. To produce sparse representations of documents, we added a MLM head with a BERT (English) vocabulary to ColBERT-X [7, 10], fine-tuned the head using the neuMARCO translation of MS MARCO [6], and then used this model to encode the document collection. To produce sparse representations of queries, we used the SPLADEv3 checkpoint [5] with no modifications. This approach bears some similarity to SPLADE-X[11], which is also a SPLADE variant with an English vocabulary, and to SPLATE [1], which trains a MLM head on top of a ColBERT checkpoint to produce sparse representations for first-stage retrieval.

Acknowledgments

This research was partially supported by the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl.

References

[1] Thibault Formal, Stéphane Clinchant, Hervé Déjean, and Carlos Lassance. 2024. Splate: Sparse late interaction retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2635–2640.

[2] Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context Autoencoder for Context Compression in a Large Language Model. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=uREj4ZuGJE

[3] John Giorgi, Luca Soldaini, Bo Wang, Gary Bader, Kyle Lo, Lucy Wang, and Arman Cohan. 2023. Open Domain Multi-document Summarization: A Comprehensive Study of Model Brittleness under Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 8177–8199. https://doi.org/10.18653/v1/2023.findings-emnlp.549

[4] Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (Eds.). Association for Computational Linguistics, Online, 874–880. https://doi.org/10.18653/v1/2021.eacl-main.74

[5] Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. SPLADE-v3: New baselines for SPLADE. arXiv:2403.06789 [cs.IR] https://arxiv.org/abs/2403.06789

[6] Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W Oard, Luca Soldaini, and Eugene Yang. 2023. Overview of the TREC 2022 NeuCLIR Track. *arXiv preprint arXiv:2304.12367* (2023).

[7] Dawn Lawrie, Eugene Yang, Douglas W Oard, and James Mayfield. 2023. Neural Approaches to Multilingual Information Retrieval. In *Proceedings of the 45th European Conference on Information Retrieval (ECIR)*. https://arxiv.org/abs/2209.01335

[8] Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2024. Learning to Compress Prompts with Gist Tokens. arXiv:2304.08467 [cs.CL] https://arxiv.org/abs/2304.08467

[9] Suraj Nair, Eugene Yang, Dawn Lawrie, Kevin Duh, Paul McNamee, Kenton Murray, James Mayfield, and Douglas W. Oard. 2022. Transfer Learning Approaches for Building Cross-Language Dense Retrieval Models. In *Proceedings of the 44th European Conference on Information Retrieval (ECIR)*. https://arxiv.org/abs/2201.08471

[10] Suraj Nair, Eugene Yang, Dawn Lawrie, Kevin Duh, Paul McNamee, Kenton Murray, James Mayfield, and Douglas W. Oard. 2022. Transfer Learning Approaches for Building Cross-Language Dense Retrieval Models. In *Proceedings of the 44th European Conference on Information Retrieval (ECIR)*. https://arxiv.org/abs/2201.08471

[11] Suraj Nair, Eugene Yang, Dawn J Lawrie, James Mayfield, and Douglas W Oard. 2022. Learning a Sparse Representation Model for Neural CLIR.. In *DESIRES*. 53–64.

[12] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL] https://arxiv.org/abs/2302.13971

[13] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models Are Zero-Shot Learners. arXiv:2109.01652 [cs.CL] https://arxiv.org/abs/2109.01652

[14] Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. RECOMP: Improving Retrieval-Augmented LMs with Compression and Selective Augmentation. arXiv:2310.04408 [cs.CL]

[15] Eugene Yang, Dawn Lawrie, and James Mayfield. 2024. Distillation for Multilingual Information Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*.

[16] Howard Yen, Tianyu Gao, and Danqi Chen. 2024. Long-Context Language Modeling with Parallel Context Encoding. In *Association for Computational Linguistics (ACL)*.