GN-TRVN: A Benchmark for Vietnamese Table Markdown Retrieval Task

Bao Loc Pham, Quoc Viet Hoang, Quy Tung Luu and Trong Thu Vo[†]

GreenNode.

*Corresponding author(s). E-mail(s): locpb@greennode.ai; viethq5@greennode.ai;
Contributing authors: tunglq2@vng.com.vn; thu@greennode.ai;
[†]These authors contributed equally to this work.

Abstract

Information retrieval often comes in plain text, lacking semi-structured text such as HTML and markdown, retrieving data that contains rich format such as table became non-trivial. In this paper, we tackle this challenge by introducing a new dataset, GreenNode Table Retrieval VN (GN-TRVN), which is collected from a massive corpus, a wide range of topics, and a longer context compared to ViQuAD2.0. To evaluate the effectiveness of our proposed dataset, we introduce two versions, M3-GN-VN and M3-GN-VN-Mixed, by fine-tuning the M3-Embedding model on this dataset. Experimental results show that our models consistently outperform the baselines, including the base model, across most evaluation criteria on various datasets such as VieQuADRetrieval, ZacLegalTextRetrieval, and GN-TRVN. In general, we release a more comprehensive dataset and two model versions that improve response performance for Vietnamese Markdown Table Retrieval.

Keywords: Vietnamese Text Embedding, Natural Language Processing, Information Retrieval, Semantic Similarity

1 Introduction

Information Retrieval is the task of converting natural language sentences or a paragraph into meaningful representations of high-dimensional vectors (embedding)[1]. With the releases and open-source of pre-trained language models, quality text embedding has been improved. There also exists a widely recognized benchmark for these text embedding models called Massive Text Embedding Benchmark[2]. This benchmark has been updated regularly by the continuing releases and publishing of individuals and organizations from all experts in the Natural Language Processing (NLP) field.

Despite the widespread popularity of text embedding, the methodology is still limited by a lack of variety. Most embedding models are focused on English and Chinese, leaving other languages dependent on multilingual models. Training a long-document retrieval system is also challenging due to the overwhelming training effort while preparing the training data. With the trend in the Retrieval-augmented generation (RAG) system, document extraction encounters a huge amount of tables, most of which are formatted as markdown, and the retrieval and fine-tuning embedding currently still lacks Table retrieval task[2].

From the issues presented above, we can see text embedding in Vietnamese is still under development due to many factors, especially a lack of data. The paper introduces GreenNode Table Retrieval, specifically designed for markdown table retrieval tasks. By utilizing LLMs, it synthesizes data from plain text into tables. Furthermore, it employs M3-Embedding [3] to fine-tune two models based on the synthesized data, thereby achieving boosted performance across several benchmarks. Our main contributions in this paper can be summarized as follows:

- This study has collected and built a dataset on Table Markdown retrieval for Vietnamese.

- To the author's knowledge, the number of research on the problem text embedding for Vietnamese is still limited, as mentioned in section 2. Therefore, this paper proposed two models that achieve better accuracy than the compared baselines. We introduce **M3-GN-VN** and **M3-GN-VN-Mixed** which are finetuned from M3-Embedding. Both models use the same dataset and fine-tuning techniques to create a wide range of choices when selecting a model for further research and deployment.

- This research proposes an approach named MTDG (Markdown-Table-data-generating pipeline) to synthesize raw text data into markdown tables using Large Language Models (LLMs).

2 Related Work

The text embedding models have been in substantial progress in research and development. The popularity of pre-trained language models where underlying semantics of sentences, and paragraphs can be expressed as meaningful matrices by utilizing transformers encoder [4]. However, the task of finding similar text pairs is time-consuming because of BERT [5] and RoBERTa [6] (which is state-of-the-art at the time) semantic textual similarity requires both sentences to be put into the network, causing a huge amount of computational overhead. The release of Sentence-BERT[1] proposed a new architecture that can extract semantic embedding that can compared using cosine-similarity. This reduces the computational overhead to query most similar text to a vector or embedding space that stores the corpus, instead of comparing each two sentences at each time. Training of an effective embedding model got critical improvement by constructing negative from Approximate Nearest Neighbor (ANN), which is select more realistic negative training sample [7]. Advance from these techniques, there are many impactful methods in versatile embedding models such as E5[8], and BGE[3].

Besides, there are also multilingual models available that were trained in Vietnamese such as E5-Multilingual [9], M3-Embedding [3], mBert[10], etc.

The PhoBERT [11] set a foundation as "the first public large-scale monolingual language models pre-trained for Vietnamese". Advancing this there are public monolingual that especially for Vietnamese text embedding such as sup-SimCSE-VietNamese-phobert-base[1], vietnamese-bi-encoder[12], vietnamese-sbert [2] , etc. Another one is the curation of training and evaluation data for Vietnamse text embeddings, e.g., Vietnamese News corpus[3], UIT-ViQuAD2.0 [13], Zalo Legal Text Retrieval[4] e.g. However, the dataset is only focused in news retrieval, wiki retrieval, and legal retrieval. In this paper, we introduce GreenNode-Table-Markdown-Retrieval-VN dataset, which focuses on markdown format table retrieval, to contribute an additional resource for Vietnamese NLP research and development, reduce the notable gap with English models, and the huge imbalance between different languages.

Despite substantial technical advancement and development from NLP communities, most of the existing text embedding models are developed only for English, where other languages like Vietnamese lagging behind.

3 Dataset Construction

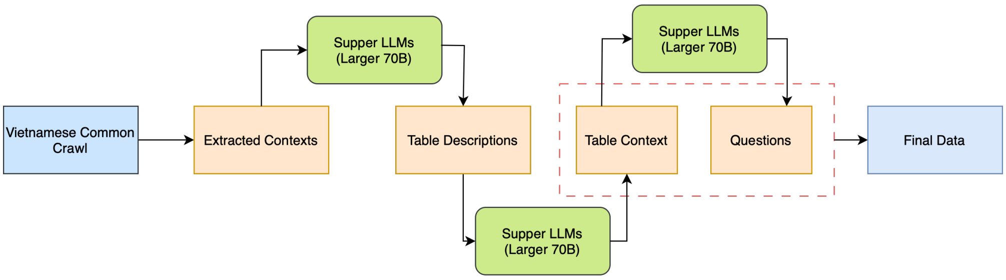


Fig. 1 Dataset construction pipeline. Using LLM to extract contexts from raw paragraphs, create table descriptions, table context and questions, saving to final data for training and evaluate.

3.1 Data Creation Process

The proposed process for creating the GN-TRVN corpus (Markdown data generator) involves five main stages: document collection, table instruction creation, table creation, table question creation, and quality assurance. Figure 1 presents an overview of the corpus creation process, with detailed descriptions provided in 3.1.

Document collection. CommonCrawl[5] is a large web dataset collected from the internet, organized into segments of web pages. We collected data from 2023 and performed preprocessing steps such as deduplication, language identification, and removal of irrelevant content, etc. After processing, we gathered a substantial amount of data, approximately 150,000 article texts in Vietnamese.

[1] https://huggingface.co/VoVanPhuc/sup-SimCSE-VietNamese-phobert-base
[2] https://huggingface.co/keepitreal/vietnamese-sbert
[3] https://github.com/binhvq/news-corpus
[4] https://challenge.zalo.ai
[5] https://commoncrawl.org

Table Instruction Creation. For each extracted-context, we create a table instruction by passing `extracted-context` to `TABLE_INSTRUCTION_CREATION`. After this step, we get a plan or a strategic to elaborate extracted-context.

```
TABLE_INSTRUCTION_CREATION = """You are a master strategist with deep analytical skills. Your task is to evaluate
      the provided information and generate a list of actionable plans. Focus on creating practical, high-impact
      strategies that can be implemented effectively.

Given the following information, analyze the context and formulate 5 strategic plans:
‘‘‘{extracted-context}‘‘‘

Your response should be concise and to the point. Provide exactly 5 plans with no additional explanations or
      elaborations.
Additional Instructions:
- Ensure that each plan is distinct and actionable.
- Prioritize effectiveness and feasibility in your plans.
- Keep the language clear and concise.

YOUR PLANs ARE:
"""
```

Table Creation. After get `plan` from 3.1, we create table from `TABLE_CREATION` prompt, with random `row` and `col` from range `4` to `8`.

```
TABLE_CREATION = """From now you are an Table Markdown Generator. You are tasked with creating a well-structured
      markdown table based on the provided strategic plan details, and then provide a brief analysis of the table.
      The analysis should include a general description, an overview of the key points, and any relevant comparisons
       or observations. The following plan details is:
‘‘‘{plan}‘‘‘

Based on the plan details above, create a markdown table with {row} rows and {col} columns. Ensure that the table is
       formatted correctly in markdown and that each cell contains relevant information. The table should be easy to
       read and well-organized.

Additional Instructions:
- Structure the content to fit naturally within the given rows and columns.
- Ensure proper alignment and formatting for markdown tables.
- Use descriptive headers if applicable, and distribute the information logically across the table.

YOUR TABLE IN MARDOWN FORMAT:
"""
```

Table Question Creation. After acquired `table_markdown` from 3.1, we passing `TABLE_QUESTION_GENERATION` to supper LLMs mention in 1 to create a dataset sample contains `table context (corpus)` and `questions (queries)`.

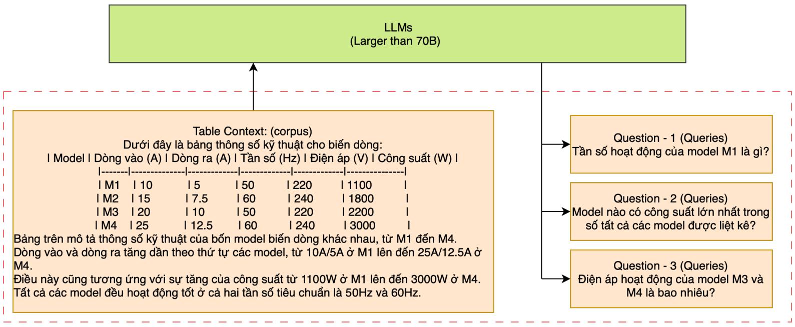
```
TABLE_QUESTION_GENERATION = """Your task is to analyze the given markdown table and generate five insightful
      questions that can be used to query or infer information from the table. These questions should vary in
      complexity, ranging from direct queries to more complex ones that require cross-referencing multiple cells or
      rows. Aim for a mix of straightforward and complex questions to fully explore the data contained in the table.
       Here is content of markdown table:
‘‘‘{table_markdown}‘‘‘

Additional Instructions:
- Ensure a balance between direct and indirect questions.
- For indirect questions, make sure they encourage deeper analysis or reasoning.
- Questions should be clear, concise, and relevant to the content of the table.
- Each question must be on a new line, using '\n' to separate them.

YOUR QUESTIONS ARE:
"""
```

Table Question Creation. After acquired `table_markdown` from 3.1, we passing `TABLE_QUESTION_GENERATION` to supper LLMs mention in 1 to create a dataset sample contains `table context (corpus)` and `questions (queries)`.

```
TRANSLATION_PROMPT = """You are a professional translator with a deep understanding of both the native language and
Vietnamese languages.
You possess a high level of proficiency in translating between these languages, ensuring that the translation
retains both the meaning and cultural nuances of the original text.
With a thorough knowledge of grammar, syntax, and colloquialisms in both languages,
you can accurately and naturally translate the following from native language to Vietnamese while maintaining its
intended tone and context.
Your work is known for its precision and attention to detail, making your translations clear and effective for
a wide audience. Translate the following from native language to Vietnamese: {{text}}
TRANSLATED VERSION IN VIETNAMESE IS: """
```

3.2 Dataset Analysis

Data after generated split into corpus and queries, follow format as MTEB Benchmark dataset [2]. With each `table context (corpus)` has multiple generated `question (queries)` from a Super LLM (see figure 2)



Dưới đây là bảng thông số kỹ thuật cho biến dòng:

```
| Model | Dòng vào (A) | Dòng ra (A) | Tần số (Hz) | Điện áp (V) | Công suất (W) |
|-------|--------------|-------------|-------------|-------------|---------------|
| M1    | 10           | 5           | 50          | 220         | 1100          |
| M2    | 15           | 7.5         | 60          | 240         | 1800          |
| M3    | 20           | 10          | 50          | 220         | 2200          |
| M4    | 25           | 12.5        | 60          | 240         | 3000          |
```

Bảng trên mô tả thông số kỹ thuật của bốn model biến dòng khác nhau, từ M1 đến M4.
Dòng vào và dòng ra tăng dần theo thứ tự các model, từ 10A/5A ở M1 lên đến 25A/12.5A ở M4.
Điều này cũng tương ứng với sự tăng của công suất từ 1100W ở M1 lên đến 3000W ở M4.
Tất cả các model đều hoạt động tốt ở cả hai tần số tiêu chuẩn là 50Hz và 60Hz.

Tần số hoạt động của model M1 là gì?

Model nào có công suất lớn nhất trong số tất cả các model được liệt kê?

Điện áp hoạt động của model M3 và M4 là bao nhiêu?

Fig. 2 Sample data. Which each table context (corpus), using LLMs larger than 70B parameters to create 3 corresponding sample question (queries).

3.2.1 Overall statistic

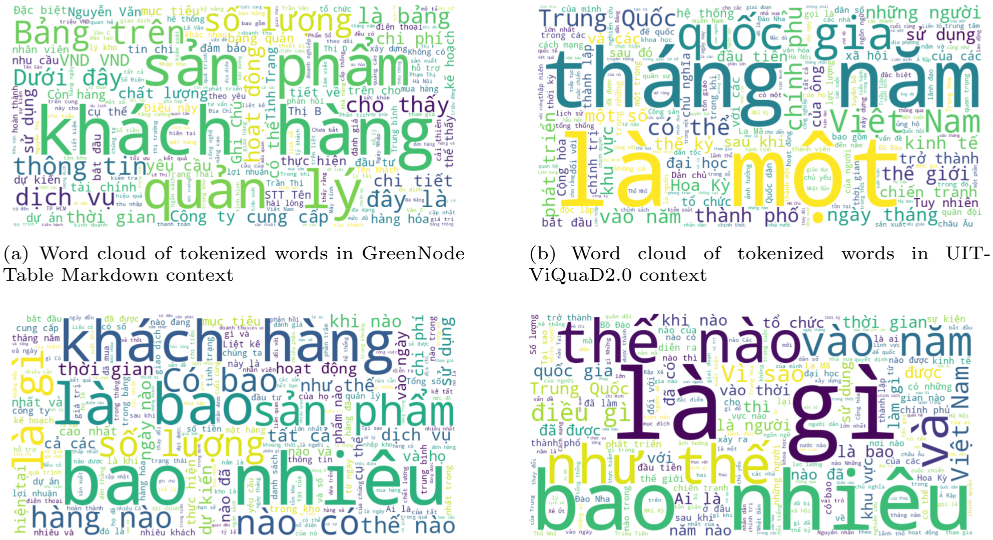
	GreenNode Table (our work)			UIT-ViQuAD2.0			
	Train	**Test**	**Total**	**Train**	**Dev**	**Test**	**Total**
Context count	35,742	8,936	44,678	4,101	557	1241	5,899
Question count	143,106	35,791	178,897	28,454	3,814	7,301	39569
Avg. context length	1,274.7	1271.2	1,274.7	853.2	815.6	840.7	847.3
Avg. question length	89.3	89.3	89.3	66.4	65.8	66.2	66.3
Vocabulary size	124,435	50,832	143,251	37,710	9,653	17,026	46,236

Table 1 Overall statistics of our dataset and UIT-ViQuAD2.0.

Inferred from Table 1, the dataset comprises 35,742 contexts from CommonCrawl[6]. We followed the methodology of the UIT-ViQuAD paper [14] to use a Python Vietnamese toolkit - pyvi [7] to segment words. In comparison to UIT-ViQuAD2.0 [13], our dataset is larger in size and vocabulary. Our dataset used more context documents, a total of 44,678 contexts compared with 5,899 passages. Additionally, the transcripts in our dataset are much longer on average, with an average length of 1,274.7 words, compared to the majority of UIT-ViQuAD's context passages ranging from 815 to 850 words. As represented in 2, context contains information and a markdown table, increasing more words than traditional text passage while a query is relatively concise.

[6] https://commoncrawl.org
[7] https://pypi.org/project/pyvi/

To further explore the distinctive vocabulary of each corpus, we created Figure 3 to display a visualization of the exclusive vocabulary in our corpus and UIT-ViQuAD2.0. In context, the UIT-ViQuAD2.0 cloud represents more formal topics, frequently occur in an informative context such as "quốc gia", "tháng năm", "thế giới", "Việt Nam". While in our corpus, we represent the context as a markdown table with its following context and refer more to statistics and management such as "bảng trên", "khách hàng", "sản phẩm", etc.



(a) Word cloud of tokenized words in GreenNode Table Markdown context

(b) Word cloud of tokenized words in UIT-ViQuaD2.0 context

(c) Word cloud of tokenized words in GreenNode Table Markdown questions (queries)

(d) Word cloud of tokenized words in UIT-ViQuaD2.0 questions (queries)

Fig. 3 Word cloud representation of GreenNode Table Markdown and UIT-ViQuAD2.0

4 Experiment Setup

4.1 Dataset for Fine-tuning

Our embedding was trained on the GreenNode Table Retrieval VN train set with 143k samples. Adapt from paper Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval (ANCE method) [7], with each positive, we create nearest approximately 7 hard negative candidates.

4.2 Fine-tuning process

Adapt from M3-Embedding [3] techniques, we employed unified fine-tuning and self-knowledge distillation to train the model on mentioned dataset. The Loss function was used is InfoNCE loss, with the formula is represented by the following:

$$\mathcal{L}_{s(\cdot)} = -\log \frac{\exp(s(q,p^*)/\tau)}{\sum_{p\in\{p^*,P'\}} \exp(s(q,p)/\tau)}. \tag{1}$$

Here, p^* and P' stand for the positive and negative samples to the query q; $s(\cdot)$ is any of the functions within $\{s_{dense}(\cdot), s_{lex}(\cdot), s_{mul}(\cdot)\}$. Parameter τ is a temperature scaling factor that controls the sharpness of the similarity distribution.

Training resource is 2×H100 and progress is over 4 epochs with 291,264 steps, batch size of 2, using learning rate 1e-5 and cosine scheduler. After tuning, we have **M3-GN-VN**, and interpolating it with the original model, **M3-Embedding**, we created **M3-GN-VN-Mixed**, which surpasses both the fine-tuned and original models in downstream task performance.

5 Evaluation

Our experiment evaluates the model's performance using three metrics: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG). These metrics measure the model's effectiveness by examining the ranking of relevant documents within the retrieval list. In essence, a higher score indicates a more effective embedding model, as it ranks the most relevant documents at the top of the retrieval list. Additionally, Section 6 provides an error analysis to assess whether the correct documents are consistently retrieved. Our embedding was evaluated on:

1. GreenNode-Table-Markdown-Retrieval-VN: test set 35.8k samples. (Our work)
2. Legal Text Retrieval Zalo 2021 challenge [8]: test set 793 samples. This widely recognized dataset was created by the ZaloAI team for the 2021 challenge, focusing on the Information Retrieval task in the legal domain.
3. UIT-VieQuADRetrieval [13] validation set 2048 samples. The early datasets for training the machine learning reading comprehension models. Where the answer answers are directly extracted from the reading passage.

The result and comparison between multilingual models, prior Vietnamese models, and our works are in table 2, 3, 4.

Since a Retrieval-Augmented Generation (RAG) system depends on both the learned knowledge of the LLM and the retrieval of documents relevant to the question, improvements in benchmark performance can directly enhance the overall accuracy and effectiveness of RAG systems [15]. However, within the scope of this paper on embedding models, we focus on evaluating the embedding model's effectiveness specifically in terms of positional ranking and correct retrieval benchmarks, as detailed in 6.

In bold and underline we show the best and second-best performances. From the results, it is evident that our fine-tuned model outperforms the majority of current state-of-the-art models, including the baseline M3-Embedding model [3]. Notably, as shown in 3, the performance in 4 demonstrates a marginal improvement over the baseline model. Although Vietnamese models from prior work are monolingual, their performance lags behind that of multilingual models, primarily due to limitations in their training data and implementation architecture. In contrast, our embedding model, trained on the proposed dataset, achieves state-of-the-art results across several Vietnamese benchmarks.

[8] https://challenge.zalo.ai

Table 2 Performance comparison of various models on GreenNodeTableRetrieval.

Model Name	MAP$_{@5}$ ↑	MRR$_{@5}$ ↑	NDCG$_{@5}$ ↑	Recall$_{@5}$ ↑	Mean ↑
		Multilingual Embedding models			
me5$_{\text{small}}$	33.75	33.75	35.68	41.49	36.17
me5$_{\text{large}}$	38.16	38.16	40.27	46.62	40.80
M3-Embedding	36.52	36.52	38.60	44.84	39.12
OpenAI-embedding-v3	30.61	30.61	32.57	38.46	33.06
		Vietnamese Embedding models (Prior Work)			
halong-embedding	32.15	32.15	34.13	40.09	34.63
sup-SimCSE-VietNamese-phobert$_{\text{base}}$	10.90	10.90	12.03	15.41	12.31
vietnamese-bi-encoder	13.61	13.61	14.63	17.68	14.89
		GreenNode-Embedding (Our Work)			
M3-GN-VN	41.85	41.85	44.15	57.05	46.23
M3-GN-VN-Mixed	**42.08**	**42.08**	**44.33**	**51.06**	**44.89**

Table 3 Performance comparison of various models on ZacLegalTextRetrieval.

Model Name	MAP$_{@5}$ ↑	MRR$_{@5}$ ↑	NDCG$_{@5}$ ↑	Recall$_{@5}$ ↑	Mean ↑
		Multilingual Embedding models			
me5$_{\text{small}}$	54.68	54.37	58.32	69.16	59.13
me5$_{\text{large}}$	60.14	59.62	64.17	76.02	64.99
M3-Embedding	69.34	68.96	73.70	86.68	74.67
OpenAI-embedding-v3	38.68	38.80	41.53	49.94	41.74
		Vietnamese Embedding models (Prior Work)			
halong-embedding	52.57	52.28	56.64	68.72	57.55
sup-SimCSE-VietNamese-phobert$_{\text{base}}$	25.15	25.07	27.81	35.79	28.46
vietnamese-bi-encoder	54.88	54.47	59.10	79.51	61.99
		GreenNode-Embedding (Our Work)			
M3-GN-VN	65.03	64.80	69.19	81.66	70.17
M3-GN-VN-Mixed	**69.75**	**69.28**	**74.01**	**86.74**	**74.95**

Table 4 Performance comparison of various models on VieQuADRetrieval.

Model Name	MAP$_{@5}$ ↑	MRR$_{@5}$ ↑	NDCG$_{@5}$ ↑	Recall$_{@5}$ ↑	Mean ↑
		Multilingual Embedding models			
me5$_{\text{small}}$	40.42	69.21	50.05	50.71	52.60
me5$_{\text{large}}$	44.18	67.81	53.04	55.86	55.22
M3-Embedding	44.08	72.28	54.07	56.01	56.61
OpenAI-embedding-v3	32.39	53.97	40.48	43.02	42.47
		Vietnamese Embedding models (Prior Work)			
halong-embedding	39.42	62.31	48.63	52.73	50.77
sup-SimCSE-VietNamese-phobert$_{\text{base}}$	20.45	35.99	26.73	29.59	28.19
vietnamese-bi-encoder	31.89	54.62	40.26	42.53	42.33
		GreenNode-Embedding (Our Work)			
M3-GN-VN	42.85	71.98	52.90	54.25	55.50
M3-GN-VN-Mixed	**44.20**	**72.64**	**54.30**	**56.30**	**56.86**

6 Error Analysis

To analyze the efficiency of different retrieval models, we use the hit rate metric, which is straightforward and easy to interpret. However, it does not account for the relative position of each matched document within the results. The hit rate@k (Top-K Hit Rate) is defined as the proportion of queries in which at least one relevant document appears within the top k retrieved documents. Q is the total number of queries, H_q^k is 1 if the query q retrieves at least one relevant item within the top k results, and 0 otherwise.

$$\text{Top-K Hit Rate (All Queries)} = \frac{1}{Q}\sum_{q=1}^{Q} H_q^k \tag{2}$$

In the hit rate metric, values are represented within the range from 0 to 1.0. In range 0 to 0.5: Low hit rate, indicating poor retrieval effectiveness, a hit rate of 0.2 suggests that only 20% of relevant documents are retrieved. In range 0.5 to 1.0:

Higher hit rate, implying better retrieval performance, a hit rate of 0.75 implies that 75% of relevant documents are retrieved, which may be acceptable depending on the application (e.g., Retrieval-Augmented Generation).

Table 5 Performance comparison of various models on GreenNodeTableRetrieval (Hit Rate).

Model Name	Hit Rate$_{@1}$ ↑	Hit Rate$_{@5}$ ↑	Hit Rate$_{@10}$ ↑	Hit Rate$_{@20}$ ↑
Multilingual Embedding models				
me5$_{small}$	38.99	53.37	59.28	65.09
me5$_{large}$	43.99	59.74	65.74	71.59
bge-m3	42.15	57.0	63.05	68.96
OpenAI-embedding-v3	-	-	-	-
Vietnamese Embedding models (Prior Work)				
halong-embedding	37.22	52.49	58.57	64.64
sup-SimCSE-VietNamese-phobert$_{base}$	14.0	24.74	30.32	36.44
vietnamese-bi-encoder	16.89	25.94	30.50	35.70
GreenNode-Embedding (Our Work)				
M3-GN-VN	**48.31**	**64.60**	**70.83**	**76.46**
M3-GN-VN-Mixed	47.94	64.24	70.43	76.14

In bold and underline we show the best and second-best performances. As shown in Table 5, our BGE-GN-Embed-VN-V1 and Mixed-V1 models outperform previous approaches in retrieving the correct documents for queries. For a large language model (LLM), using top-k values of 5 and 10 is effective for retrieving relevant documents, as increasing k beyond 10 becomes impractical due to LLMs' limited ability to handle long contexts. While LLMs can process hundreds of thousands of tokens, their understanding diminishes with longer sequences[4]. Given the importance of positional proximity for reasoning, ranking relevant documents closer to the query is crucial for accuracy and efficiency. Our evaluation metrics in 5 account for this, with higher scores indicating more effective retrieval.

7 Limitations

The evaluation uses only Dense embedding for compatibility with the usage of Sentence Transformer [1] and for a benchmarking model for MTEB leaderboard [2]. Other retrieval types e.g. sparse vector, and multi-vector retrieval is left for further research. The dataset GreenNode-Table-Retrieval-VN is focused on markdown format, leftover HTML format, and semi-structure data to further develop.

This paper only covers a small subset of tasks in Vietnamese retrieval. Lacking of more complex data type retrieval e.g. code retrieval, news retrieval, medical document retrieval, etc.

8 Conclusion

In our error analysis, we compared our work with prior studies and concluded that the pipeline for synthesizing data 3.1, model fine-tuning, and mixing 4 create a better embedding for specific tasks. When researchers or developers want to improve model performance on a new task, they can follow our pipeline approach and further extend to a larger embedding model (LLM-based embedding model).

Although there is a performance improvement in table markdown retrieval and other tasks, there remains room for further enhancement. Implementing a re-ranking model or using improved embeddings could further boost retrieval accuracy by prioritizing relevant documents at the top of the results, which is particularly beneficial for a Retrieval-Augmented Generation (RAG) system.

Acknowledgements. This research is supported by GreenNode-AI, experiments was performed on their generous sponsored GPU H100.

References

[1] Reimers, N.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)

[2] Muennighoff, N., Tazi, N., Magne, L., Reimers, N.: Mteb: Massive text embedding benchmark. arXiv preprint arXiv:2210.07316 (2022)

[3] Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., Liu, Z.: Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv preprint arXiv:2402.03216 (2024)

[4] Vaswani, A.: Attention is all you need. Advances in Neural Information Processing Systems (2017)

[5] Devlin, J.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[6] Liu, Y.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 **364** (2019)

[7] Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P., Ahmed, J., Overwijk, A.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. arXiv preprint arXiv:2007.00808 (2020)

[8] 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)

[9] Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., Wei, F.: Multilingual e5 text embeddings: A technical report. arXiv preprint arXiv:2402.05672 (2024)

[10] Pires, T.: How multilingual is multilingual bert. arXiv preprint arXiv:1906.01502 (2019)

[11] Nguyen, D.Q., Nguyen, A.T.: Phobert: Pre-trained language models for vietnamese. arXiv preprint arXiv:2003.00744 (2020)

[12] Duc, N.Q., Son, L.H., Nhan, N.D., Minh, N.D.N., Huong, L.T., Sang, D.V.: Towards comprehensive vietnamese retrieval-augmented generation and large language models. arXiv preprint arXiv:2403.01616 (2024)

[13] Van Kiet, N., Son, T.Q., Luan, N.T., Van Tin, H., Son, L.T., Ngan, N.L.T.: Vlsp 2021-vimrc challenge: Vietnamese machine reading comprehension. VNU Journal of Science: Computer Science and Communication Engineering **38**(2) (2022)

[14] Van Nguyen, K., Nguyen, D.-V., Nguyen, A.G.-T., Nguyen, N.L.-T.: A vietnamese dataset for evaluating machine reading comprehension. arXiv preprint arXiv:2009.14725 (2020)

[15] Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., Grave, E.: Atlas: Few-shot learning with retrieval augmented language models. Journal of Machine Learning Research **24**(251), 1–43 (2023)