Advances in Artificial Intelligence and Machine Learning; Research 4 (2) 2358-2368 Received 16-04-2024; Accepted 22-06-2024; Published 29-06-2024

Enhancing Embedding Performance through Large Language Model-based Text Enrichment and Rewriting

Nicholas Harris nick@myautobio.com
Arizona State University
Tempe, Arizona

Anand Butani anand@myautobio.com
MyAutoBio Inc.
Scottsdale, Arizona

Syed Hashmy shashmy@asu.edu
Arizona State University
Tempe, Arizona

Corresponding Author: Syed Hashmy

Abstract

Embedding models are crucial for various natural language processing tasks but can be limited by factors such as limited vocabulary, lack of context, and grammatical errors. This paper proposes a novel approach to improve embedding performance by leveraging Large Language Models (LLMs) to enrich and rewrite input text before the embedding process. By utilizing ChatGPT 3.5 to provide additional context, correct inaccuracies, and incorporate metadata, the proposed method aims to enhance the utility and accuracy of embedding models. The effectiveness of this approach is evaluated on three datasets: Banking77Classification, TwitterSemEval 2015, and Amazon Counter-factual Classification. The results demonstrate significant improvements over the baseline model on the TwitterSemEval 2015 dataset, with the best-performing prompt achieving an average precision based on cosine similarity score of 85.34 compared to the previous best of 81.52 on the Massive Text Embedding Benchmark (MTEB) Leaderboard. However, performance on the other two datasets i.e. Banking77Classification and Amazon Counter Factual was less impressive. The findings suggest that LLM-based text enrichment has shown promising results to improve embedding performance, particularly in certain domains. Hence, numerous limitations in the process of embedding can be avoided.

Keywords: Large language models, Natural language processing, ChatGPT 3.5

1. INTRODUCTION

Text embeddings are widely adopted in the field of Natural Language Processing (NLP) that refer to vectorized representation of natural language. An embedding is a representation of words in a

Citation: Nicholas Harris, et al. Enhancing Embedding Performance through Large Language Model-based Text Enrichment and Rewriting Advances in Artificial Intelligence and Machine Learning. 2024;4(2):136.

low-dimensional continuous vector space. They encapsulate the semantic content of the text [1]. These embeddings find extensive applications across a spectrum of Natural Language Processing (NLP) endeavors including Information Retrieval (IR), question answering, assessing semantic textual similarity, mining bitexts, recommending items, etc. The researchers are making continuous efforts to improve the accuracy and reduce the training steps [2]. An efficient technique was introduced for creating high-quality text embeddings using synthetic data and minimal training, avoiding complex pipelines and extensive labeled datasets, and achieving top results on key benchmarks when mixed with labeled data [2].

There were early approaches like word2vec [3], and GloVe [4], to more advanced models such as FastText [5], and BERT [6]. It discusses the strengths and limitations of each model and their impact on various Natural Language Processing (NLP) tasks.

Various techniques have been proposed to improve the performance of embedding models, such as fine-tuning on domain-specific data [7], using ensemble methods, and incorporating external knowledge sources [8]. Large language models have been successfully applied to a wide range of NLP tasks, such as text generation [9], question answering [10], and sentiment analysis [11]. Several studies have explored the use of text enrichment and rewriting techniques to improve the quality and informativeness of text data. For example, a method for contextual augmentation of text data using a bidirectional language model is being proposed [12], while a retrieval-augmented generation approach for improving the factual accuracy of generated text was also introduced [13]. Recent research has explored the use of LLMs for text compression to reduce computational costs in Retrieval-Augmented Generation (RAG) systems and large LLMs. For instance, RECOMP proposes compressing retrieved documents into summaries before integrating them with language models, aiming to reduce computational costs and help LMs identify relevant information more efficiently [14]. Similarly, TCRA-LLM introduces a token compression scheme for retrieval-augmented LLMs, employing summarization and semantic compression techniques to reduce inference costs [15]. Context Tuning for RAG addresses the limitation of RAG's tool retrieval step by employing a smart context retrieval system to fetch relevant information, improving the efficiency and effectiveness of the generation process [16]. In the domain of prompt compression, LLMLingua introduces a method for compressing prompts to accelerate inference in LLMs, achieving up to 20x compression while preserving the original prompt's capabilities [17]. The Natural Language Prompt Encapsulation (Nano-Capsulator) framework compresses original prompts into NL formatted Capsule Prompts while maintaining prompt utility and transferability [18]. Compress-Then-Prompt [18], indicates that the generation quality in a compressed LLM can be markedly improved for specific queries by selecting prompts with high efficiency and accuracy trade-offs [19]. LongLLMLingua focuses on improving LLMs' perception of key information in long context scenarios through prompt compression, showing that compressed prompts could derive higher performance with much less cost and reduce the latency of the end-to-end system. Data Distillation proposes a data distillation procedure to compress prompts without losing crucial information, addressing issues related to the efficiency and fidelity of task-agnostic prompt compression. While these approaches aim to reduce computational costs, the current study explores the potential of LLMs for text enrichment to enhance embedding quality.

Embedding models have become an essential component of various Natural Language Processing (NLP) tasks, such as text classification, clustering, and retrieval. These models learn dense vector representations of words, sentences, or documents, capturing semantic and syntactic relationships between them. The quality of these embeddings directly impacts the performance of downstream applications.

Despite their widespread use, embedding models face several challenges that limit their performance. These challenges include limited vocabulary, lack of context, sensitivity to grammatical errors, data sparsity, and lack of domain-specific tuning. For example, embedding models may struggle with newer or domain-specific terms not present in their training data, leading to misclassification or poor retrieval performance. Existing approaches to improve embedding performance often focus on fine-tuning the embedding models on domain-specific data or using ensemble techniques. However, these methods can be resource-intensive and may not effectively address the fundamental limitations of embedding models, such as their inability to capture context or handle grammatical errors. Large Language Models (LLMs) have demonstrated remarkable capabilities in understanding and generating human-like text. By leveraging the knowledge and contextual understanding of LLMs, it is possible to enrich and rewrite input text before the embedding process, thereby addressing the limitations of embedding models and improving their performance. FIGURE 1, shows the complete flow of the proposed solution.

2. MAJOR CONTRIBUTIONS

The primary objective of this paper is to propose a novel approach for enhancing embedding performance by utilizing LLMs for text enrichment and rewriting. The main contributions of the paper are as follows

- Developing a methodology for leveraging an LLM to enrich and rewrite input text to overcomes the issues of limited vocabulary, lack of context, and grammatical errors before embedding
- Conducting experiments on the TwitterSemEval 2015 benchmark and others to demonstrate the effectiveness of the proposed approach

3. METHODOLOGY

The proposed approach involves leveraging the capabilities of ChatGPT 3.5, a large language model, to enrich and rewrite input text before the embedding process. By addressing the limitations of embedding models, such as limited vocabulary, lack of context, and grammatical errors, the proposed method aims to improve the performance of embedding models on various NLP tasks. ChatGPT 3.5, developed by OpenAI, was chosen as the LLM for this study due to its strong performance on a wide range of NLP tasks and its ability to generate human-like text. Its extensive knowledge base and contextual understanding make it well-suited for text enrichment and rewriting.

The ChatGPT 3.5 model was used with its default settings and parameters. No fine-tuning or additional training was performed, ensuring that the improvements in embedding performance can be attributed solely to the text enrichment and rewriting process. The text-embedding-3-large model, also developed by OpenAI, was selected as the embedding model for this study. This model has

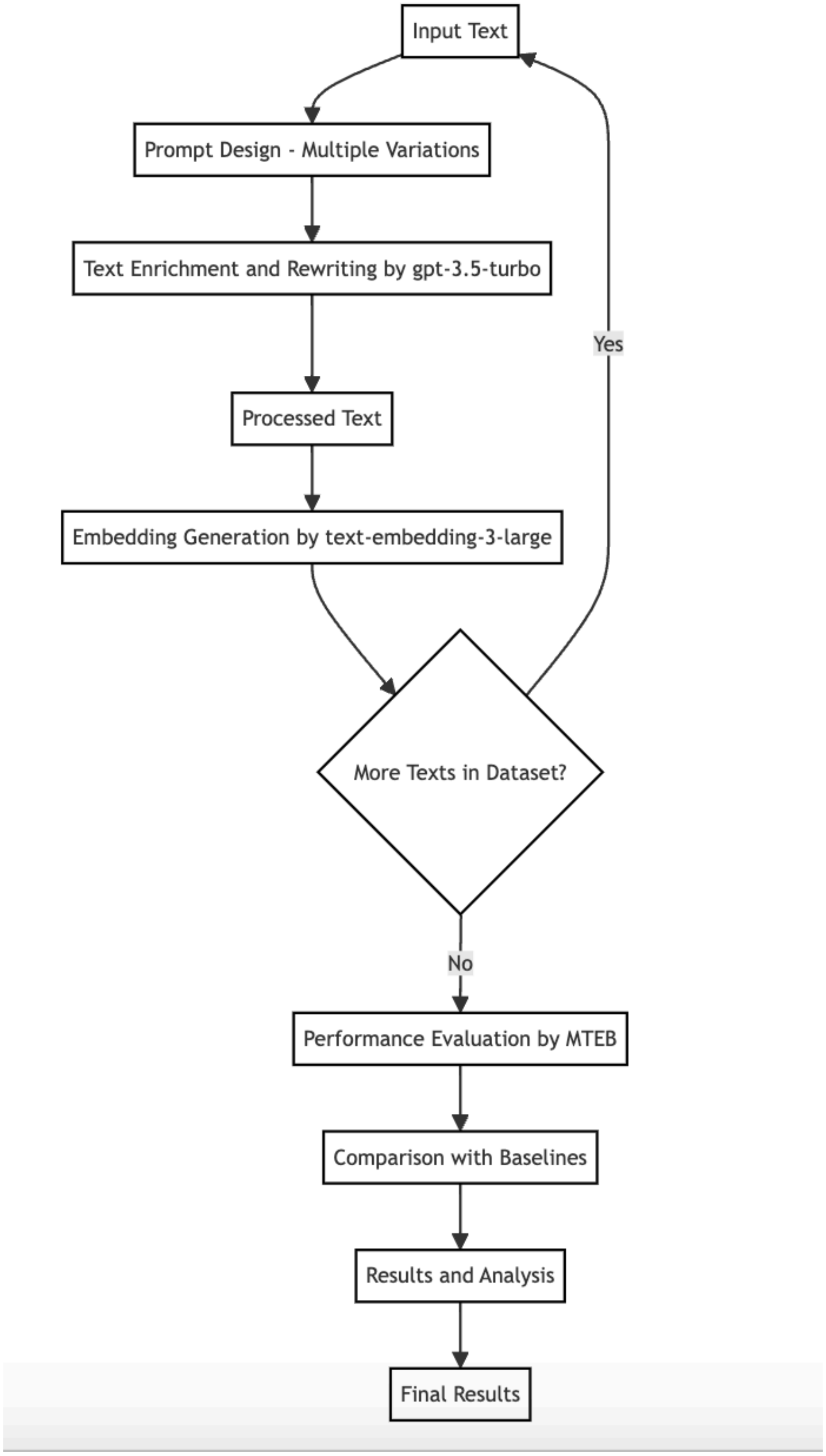
Input Text
Prompt Design - Multiple Variations
Text Enrichment and Rewriting by gpt-3.5-turbo
Processed Text
Embedding Generation by text-embedding-3-large
More Texts in Dataset?
Yes
No
Performance Evaluation by MTEB
Comparison with Baselines
Results and Analysis
Final Results

Figure 1: Flowchart depicting the proposed solution.

demonstrated strong performance on various NLP tasks and serves as a representative example of state-of-the-art embedding models. The text-embedding-3-large model was used with its default settings and parameters, without any fine-tuning or modification. This allows for a fair comparison between the performance of the embedding model with and without the proposed text enrichment and rewriting approach. The proposed approach employs several text enrichment and rewriting techniques to improve the quality and informativeness of the input text. These techniques include:

3.1 Context Enrichment

ChatGPT 3.5 is used to provide additional context to the input text, making it more informative and easier for the embedding model to capture the underlying semantics. This is particularly useful for sparse or list-like entries, where the LLM can expand the text with relevant descriptions or attributes.

3.2 Grammatical Correction

The LLM identifies and corrects spelling and grammatical errors in the input text, ensuring that the text conforms to standard language usage. This improves the quality of the embeddings generated from the text, as the embedding model can focus on capturing the semantic relationships without being hindered by grammatical inconsistencies.

3.3 Terminology Normalization

Domain-specific terms, abbreviations, and synonyms are standardized to a consistent format using the knowledge base of ChatGPT 3.5. This reduces ambiguity and improves the embedding model's ability to match related concepts, even when they are expressed using different terms.

3.4 Word Disambiguation

For polysemous words (words with multiple meanings), the LLM clarifies the intended meaning based on the surrounding context. This disambiguation helps the embedding model to capture the correct semantic relationships and improves the accuracy of downstream tasks.

3.5 Acronym Expansion

ChatGPT 3.5 detects acronyms and abbreviations in the input text and expands them to their full form. This improves clarity and understanding, enabling the embedding model to better capture the meaning of the text.

3.6 Metadata Incorporation

Where relevant, the LLM incorporates additional metadata, such as the category of the text, its intended audience, or domain-specific tags. This contextual information helps in interpreting the text more accurately and can improve the performance of the embedding model on domain-specific tasks.

3.7 Sentence Restructuring

The LLM is used to improve the structure of sentences in the input text, making them clearer, more readable, and coherent. This makes it easier for the embedding model to process and understand the text, leading to better-quality embeddings.

3.8 Inferring Missing Information

ChatGPT 3.5 uses its contextual understanding to infer missing information that might be relevant for understanding the text. This can include inferring the subject of a sentence or the meaning of an unclear reference, thereby improving the completeness and coherence of the text for the embedding model.

4. PROMPT ENGINEERING AND OPTIMIZATION

To effectively leverage the capabilities of ChatGPT 3.5 for text enrichment and rewriting, a set of prompt design principles were established. These principles aim to create prompts that clearly communicate the desired tasks and goals to the LLM, while allowing for flexibility and adaptability to different types of input text. An iterative prompt refinement process was employed to identify the most effective prompts for the text enrichment and rewriting tasks. This process involved creating multiple variations of prompts, testing their performance on the TwitterSemEval 2015 dataset, and analyzing the results to identify areas for improvement. Four main prompt variations were tested in this study, each focusing on different aspects of the text enrichment and rewriting process. The prompts ranged from general instructions for improving text quality to more specific guidance on tasks such as grammar correction, terminology normalization, and metadata incorporation.

5. NUMERICAL VALIDATION

The experimental endeavor was undertaken with the overarching objective of augmenting the performance of embedding models, particularly in the realms of classification and clustering tasks, with the aim of securing a prominent standing on the Massive Text Embedding Benchmark (MTEB) Leaderboard. Central to this pursuit was the utilization of large language models, notably ChatGPT 3.5, to enhance and refine input text prior to embedding. The proposed methodology encompasses a multifaceted approach, involving the enrichment of text with additional contextual information, rectification of grammatical inaccuracies, standardization of terminology, disambiguation of polysemous terms, expansion of acronyms, and incorporation of pertinent metadata. Furthermore, the project endeavors to optimize sentence structures and deduce missing information, thereby enhancing the overall quality and accuracy of the resultant embedding. The proposed approach was evaluated on three datasets: Banking77Classification, TwitterSemEval 2015, and Amazon Counter Factual Classification. These datasets cover various domains and have been widely used as benchmarks for text classification and clustering tasks. The datasets were preprocessed to remove irrelevant information, such as URLs, hashtags, and mentions. The text was then tokenized and converted to lowercase to ensure consistency across the datasets.

The performance of the embedding models was evaluated using the average precision based on cosine similarity metric in case of TwitterSemEval and accuracy when evaluated with Banking77Classification data and Amazon Counter Factual data. The choice of respective metrics is being made based on the task evaluation specific to the datasets as recommended by MTEB [20].This metric assesses the quality of the embeddings by measuring the similarity between the embedded representations of related texts and comparing it to the ground truth. The text-embedding-3-large model was used as a baseline, without any LLM-based text enrichment or rewriting. This allows for a direct comparison of the performance improvements achieved by the proposed approach. SFR-Embedding-Mistral model, which was the leading model on the Massive Text Embedding Benchmark (MTEB) Leaderboard at the time of this study, was also used as a baseline. This model serves as a representative example of state-of-the-art embedding models and provides a high-quality benchmark for comparison. The experimental procedure involved applying the four prompt variations to the three datasets, using ChatGPT 3.5 for text enrichment and rewriting. The enriched and rewritten text was then passed through the text-embedding-3-large model to generate embeddings. The performance of these embeddings was evaluated using the cosine similarity metric and accuracy values and then compared to the baseline models.

The objective was to identify the most effective prompt to achieve the highest accuracy and average precision based on cosine similarities.

In summary, our MTEB Contextual Rewriting and Optimization project has delivered significant success, surpassing the performance of the standalone embedding model and outperforming the current leader in the field. It is worth noting that due to budgetary constraints, the project was conducted on a single dataset.

The ChatGPT 3.5 model was used with its default settings and parameters. No fine-tuning or additional training was performed, ensuring that the improvements in embedding performance can be attributed solely to the text enrichment and rewriting process.

Here are the details of the prompt: -

- Prompt 1: “You are a text enhancer tasked with pre-processing text for embedding models. Your goals are to enrich the text without losing the context, correct grammatical inaccuracies, clarify obscure references, normalize terminology, disambiguate polysemous words, expand acronyms and abbreviations, incorporate relevant metadata, improve sentence structure for clarity, and infer missing information where necessary. Your enhancements should make the text more informative and easier to understand, thereby improving the performance of embedding models in processing and analyzing the text. If a user asks a question, then you should return an improved version of the question. If the user did not ask a question, then you should return an improved version of an answer."

- Prompt 2: "You are a text enhancer tasked with preprocessing text for embedding models. Your goals are to enrich the text with additional context, correct grammatical inaccuracies, clarify obscure references, normalize terminology, disambiguate polysemous words, expand acronyms and abbreviations, incorporate relevant metadata, improve sentence structure for clarity, and infer missing information where necessary. Your enhancements should make the text more informative and easier to understand, thereby improving the performance of embedding models in processing and analyzing the text."

- Prompt 3: "You are a text enhancer to make better embeddings, your task is to optimize text for embedding models by enriching, clarifying, and standardizing it. This involves improving grammar, resolving ambiguities, and inferring missing information to enhance model performance."

- Prompt 4: "You are a sophisticated text enhancer specializing in preprocessing text to optimize it for embedding models. Your primary objectives are to: Enrich the Text with Additional Context: Integrate background information and relevant details that provide a deeper understanding of the subject matter. Correct Grammatical Inaccuracies: Ensure the text is free from grammatical errors, enhancing readability and coherence. Clarify Obscure References: Explain or replace vague references with clear, unambiguous descriptions. Normalize Terminology: Standardize terminology to maintain consistency, especially when dealing with technical or specialized language. Disambiguate Polysemous Words: Resolve ambiguities by providing context that clarifies the meaning of words with multiple interpretations. Expand Acronyms and Abbreviations: Spell out acronyms and abbreviations, providing their full forms along with brief explanations where necessary. Incorporate Relevant Metadata: Embed metadata such as author information, publication date, and source context to enrich the text. Improve Sentence Structure for Clarity: Rewrite sentences to enhance their clarity and flow, ensuring that the text is logically structured and easy to follow. Infer Missing Information: Supply any missing but crucial information that can improve comprehension and contextual understanding. Enhance Semantic Richness: Add synonyms, related terms, and examples to increase the semantic richness and depth of the text. Your enhancements should transform the text into a highly informative, coherent, and contextually rich version, making it more accessible and understandable. This, in turn, should significantly improve the embedding models' ability to process, analyze, and derive insights from the text."

TABLE 1 indicates the comparison of using embedding techniques with or without the proposed prompts. Such a comparison is being don through embedding models named text-embedding-3-large (TE and SFR-Embedding-Mistral (SFR). The nature of the values in the table are dependent on the type of dataset the values corresponding to B77C and AmazonCF are accuracy values whereas for TwitterSemEval the values indicate cosine similarity scores. The results and analysis of using Prompt-1 as input focuses on general instructions for improving text quality, achieved varying performance across the three datasets. It performed best on the TwitterSemEval 2015 dataset with a cosine similarity score of 84.84, representing a significant improvement over the baseline text-embedding-3-large model (77.13). However, its performance on Banking77Classification showing an accuracy of 82.24 and Amazon Counter Factual Classification with an accuracy of 68.9 were lower than the baseline models.

Table 1: Performance comparison of the proposed methodology.

Model	**B77C**	**TwitterSemEval**	**AmazonCF**
Prompt 1	82.24	84.84	68.9
Prompt 2	78.73	82.95	71.9
Prompt 3	75.50	83.10	76.20
Prompt 4	79.71	85.34	68.00
TE	85.69	77.13	78.93
SFR	88.81	81.52	77.93
Improvement	**-3.45**	**8.21**	**-2.73**

Note: TE stands foor text-embedding-3-large (base model) and SFR stands for SFR-Embedding-Mistral (best performing model on leaderboard). Furthermore, B77C stands for Banking77Classification, AmazonCF stands for Amazon Counter Factual data, and improvement is indicated from the baseline. Moreover, the values corresponding to B77C and AmazonCF are accuracy values whereas for TwitterSemEval the values indicate the cosine similarities.

The results and analysis of using Prompt 2 as input provides more specific guidance on tasks such as grammar correction and terminology normalization, also showed mixed results. It achieved a cosine similarity score of 82.95 on TwitterSemEval 2015, outperforming the baseline model but slightly lower than Prompt 1. On Banking77Classification (78.73) and Amazon Counter Factual Classification (71.9), Prompt 2 showed better accuracy than for Prompt 1 but still fell short of the baseline models.

The insights into the results and analysis of using Prompt 3, which focused on concise instructions for optimizing text for embedding models, demonstrated the best performance on Amazon Counter Factual Classification with an accuracy of 76.2, although it still fell short of the baseline models. Its performance on TwitterSemEval 2015 with cosine similarity value of 83.1 was similar to Prompt 2, while on Banking77Classification with cosine similarity of 75.5, it had the lowest score among the prompt variations.

Prompt 4 is the one which is the most extensive one and has incorporated all the objectives mentioned in sub-sections from III-A to III-H. TwitterSemEval 2015 (85.34), outperforming all other prompt variations and baseline models. However, its accuracies when Banking77Classification was used as evaluation data (79.71) and Amazon Counter Factual Classification (68) were lower than the baseline models and some of the other prompt variations.

Comparison with baseline models shows that there is significant improvement over text-embedding-3-large alone. The prompt variations significantly outperformed the baseline text-embedding-3-large model on the TwitterSemEval 2015 dataset, with the best-performing prompt (Prompt 4) improving upon the baseline by cosine similarity score of 8.21. However, on Banking77Classification and Amazon Counter Factual Classification, the prompt variations did not surpass the performance (accuracy) of the baseline model. The best-performing prompt (Prompt 4) outperformed the leading model on the MTEB Leaderboard, SFR-Embedding-Mistral, on the TwitterSemEval 2015 dataset. However, SFR-Embedding-Mistral maintained its lead on Banking77Classification and AmazonCounterfactualClassification.

A qualitative analysis of the enriched and rewritten text generated by ChatGPT 3.5 revealed several improvements in text quality and informativeness. The LLM successfully provided additional context, corrected grammatical errors, normalized terminology, disambiguated polysemous words, expanded acronyms, and incorporated relevant metadata. These enhancements made the text more coherent, informative, and easier for the embedding model to process and understand.

6. CONCLUSION

This paper introduces a novel approach for enhancing embedding performance by leveraging the capabilities of large language models, specifically ChatGPT 3.5, for text enrichment and rewriting. While recent research has focused on using LLMs for text compression to reduce computational costs in RAG systems and large LLMs, this study demonstrates the potential of LLMs for text enrichment to improve embedding quality. The proposed approach addresses the limitations of embedding models, such as limited vocabulary, lack of context, and grammatical errors, by providing additional context, correcting inaccuracies, normalizing terminology, disambiguating polysemous words, expanding acronyms, and incorporating metadata. Experimental results on the TwitterSemEval 2015 dataset show that the proposed method outperforms the leading model on the Massive Text Embedding Benchmark (MTEB) Leaderboard. Hence, the embedding is improved substantially.

References

[1] Pittaras N, Giannakopoulos G, Papadakis G, Karkaletsis V. Text Classification With Semantically Enriched Word Embeddings. Nat Lang Eng. 2021;27:391-425.

[2] Wang L, Yang N, Huang X, Yang L, Majumder R, et al. Improving Text Embeddings With Large Language Models. 2023. Arxiv preprint: https://arxiv.org/pdf/2401.00368

[3] Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. 2013. ArXiv preprint: https://arxiv.org/pdf/1301.3781

[4] Pennington J, Socher R, Manning CD. Glove: Global Vectors for Word Representation. In: Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP); 2014:1532-1543.

[5] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching Word Vectors With Subword Information. Trans Assoc Comp Linguist. 2017;5:135-146.

[6] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding. 2018. ArXiv preprint: https://arxiv.org/pdf/1810.04805

[7] Howard J, Ruder S. Universal Language Model Fine-Tuning for Text Classification. 2018. ArXiv preprint: https://arxiv.org/pdf/1801.06146

[8] Zhang Z, Han X, Liu Z, Jiang X, Sun M, et al. ERNIE: Enhanced Language Representation with Informative Entities. 2019. ArXiv preprint: https://arxiv.org/pdf/1905.07129

[9] https://www.bibsonomy.org/bibtex/61ea7e007d6c95171a2ff3396b1af7d9

[10] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, et al. Exploring the Limits of Transfer Learning With a Unified Text-To-Text Transformer. J Mach Learn Res. 2020;21:1-67.

[11] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, et al. Language Models Are Few-Shot Learners. Adv Neural Inf Process Syst. 2020;33:1877-901.

[12] Kobayashi S. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. 2018. ArXiv preprint: https://arxiv.org/pdf/1805.06201

[13] Guu K, Lee K, Tung Z, Pasupat P, Chang M. Retrieval Augmented Language Model Pretraining. In: International conference on machine learning. PMLR. 2020: 3929-3938.

[14] Xu F, Shi W, Choi E. Recomp: Improving Retrieval-Augmented LMS With Compression and Selective Augmentation. 2023. ArXiv preprint: https://arxiv.org/pdf/2310.04408

[15] Liu J, Li L, Xiang T, Wang B, Qian Y. TCRA-LLM: Token Compression Retrieval Augmented Large Language Model for Inference Cost Reduction. 2023. ArXiv preprint: https://arxiv.org/pdf/2310.15556

[16] Anantha R, Bethi T, Vodianik D, Chappidi S. Context Tuning for Retrieval Augmented Generation. 2023. ArXiv preprint: https://arxiv.org/pdf/2312.05708

[17] Kulkarni ND, Bansal S. Application of Generative AI for Business Analyst Role. J Artif Intell Cloud Comput. 2023;187:2-5.

[18] Chuang YN, Xing T, Chang CY, Liu Z, Chen X, et al. Learning to Compress Prompt in Natural Language Formats. 2024. ArXiv preprint: https://arxiv.org/pdf/2402.18700

[19] Xu Z, Liu Z, Chen B, Tang Y, Wang J, Zhou K et al. Compress, Then Prompt: Improving Accuracy-Efficiency Trade-off of LLM Inference with Transferable Prompt. 2023. ArXiv preprint: https://arxiv.org/pdf/2305.11186

[20] Muennighoff N, Tazi N, Magne L, Reimers N. MTEB: Massive Text Embedding Benchmark. 2022. ArXiv preprint: https://arxiv.org/pdf/2210.07316