One of the Approaches to Natural Language Prompting for the Large Language Model Using the University of Tyumen Document Storage

Timur Damirovich Nizamov[1] [1], Daniil Davidovich Dekanadze[1] [2] ,
Marina Sergeevna Vorobeva[2] [3] and Anna Valerevna Glazkova[2] [4]
[1]School of Computer Sciences, University of Tyumen, 6 Volodarskogo Street, Tyumen, Russia
[2]Department of Software, University of Tyumen, 6 Volodarskogo Street, Tyumen, Russia
mail@nizamovtimur.ru, dekanadze2103@gmail.com, m.s.vorobeva@utmn.ru, a.v.glazkova@utmn.ru

Keywords: Retrieval augmented generation, chatbot, virtual assistant for students, wiki system, Sentence Transformer, large language model

Abstract: The following paper explores the application of Retrieval augmented generation (RAG) for answering questions posed by students at the University of Tyumen using the large language model (LLM) and the corporate wiki system. To enhance the precision of document retrieval for question-answering purposes, the sentence embedding model is fine-tuned on the set of questions that has been generated by the LLM to corporate university documents. The results demonstrate improved performance compared to using embeddings from a third party LLM service. As a result the microservice for answering students' questions has been developed and tested for using it throughout the UTMN Virtual Assistant.

1 INTRODUCTION

At the University of Tyumen (UTMN) work is underway to digitize student support, including increasing activity on social media. For example, the Student Affairs Office has launched chatbots in VK and Telegram that provide reference information to students by pre-prepared scheme. However, freshmen and foreign students might not know which section contains information of interest, furthermore, the amount of instructions and regulations is increasing every year.

For providing reference information to students online chatbots are being developed and implemented at Russian and foreign universities (Sukhanova and Vezhelis, 2022). The researcher from the University of Hagen in the article (Hartrumpf, 2004) for question-answering system development examined the approach based on semantic networks.

Large language models (LLM) development based on Transformers (Vaswani et al., 2017) opens a wide range of possibilities for making various services including chatbots, virtual assistants, question-answering systems (Galeev and Panishchev, 2022). Retrieval augmented generation (RAG) is being used for making LLM providing actual and reliable information (Lewis et al., 2020). Prompt to LLM is being supplemented with the text relevant to the question using the proximity of embeddings. Furthermore, because of model token limits text chunking must be provided.

RAG is possible to use in various domains. For example, in papers (Al Ghadban et al., 2023) and (Ke et al., 2024) possibilities of using RAG for increasing of medical service provision quality are highlighted, in this case ready solutions to get embeddings are used during document extraction stage: pretrained on the medicine text corpus ClinicalBERT model (Wang et al., 2023) or API for getting embeddings from LLM services.

The objective is development of the Retrieval augmented generation system for answering UTMN students' questions by instructions and regulations published in corporate wiki system. There is the hypothesis that fine-tuning of the sentence embedding model on the set of questions to corporate documents that has been generated by the LLM will

[1] https://orcid.org/0009-0006-3246-6712
[2] https://orcid.org/0009-0004-3343-2632
[3] https://orcid.org/0000-0002-1508-4089
[4] https://orcid.org/0000-0001-8409-6457

increase the documents retrieving accuracy for answering questions compared to using embeddings from a LLM service.

Given the question $Q = \{q_1, q_2, ..., q_n\}$ — the set of tokens in question, a set of document chunks $D = \{d_1, d_2, .. d_m\}$, $\forall\, d_i \in D$: $d_i = \{d_{i1}, d_{i2}, ..., d_{ik}\}$ — the set of tokens in a chunk. Furthermore, $|Q| \leq K$, $|d_i| \leq K$, where K — the feature extraction model token limit.

Generate the answer $A = \{a_1, a_2, ..., a_l\}$ — the set of tokens depending on the question Q and the chunk $d_j \in D$, $j = argmax_{p=1..m}\ similarity(Q, d_p)$, where *similarity* — the proximity function (measure). Furthermore $|Q| + |d_j| + |A| \leq L$, where L — the model token limit.

Thereby, it is necessary to solve two subtasks:

1. retrieving the chunk d_j that is nearest to the question Q;
2. generating the answer A to the question Q by the chunk d_j.

For retrieving the chunk d_j it is required to get embeddings for Q and every element of the set D. Furthermore, it is important to consider the feature extraction model limit K, so documents violating this limit are needed to go through chunking. The proximity measure similarity is cosine similarity between embeddings of token sets that is not affected by characteristic for natural language texts data sparsity.

When the chunk d_j is found, it is necessary to generate the answer A to the answer Q. Answer generating is carried out by the large language model. Furthermore, it is important to consider the generative model token limit L.

2 MATERIALS AND METHODS

Retrieval augmented generation (see Figure 1) assumes the presence of a tokenizer for document chunking, the document from the set D chunks feature extraction method and the question Q for getting embeddings, vector storage for the set D, large language model for generating the answer A and the prompt.

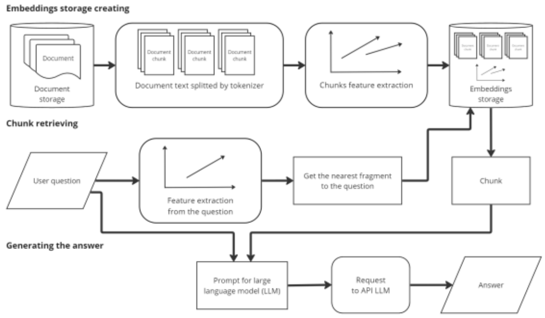

Figure 1: Flow diagram of the Retrieval Augmented Generation.

The following methods for getting embeddings are examined: TF-IDF (Sparck Jones, 1972) based on bag of words, Doc2vec (Mikolov and Le, 2014) developed based on Word2vec (Mikolov et al., 2013), Transformer-based Sentence BERT (Reimers and Gurevych, 2019), taking into account the context of words at the sentence level.

To assess the compliance of found or generated with reference texts metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) are being used. Initially developed for assessment of machine translation quality, BLEU measures similarity between reference text and generated text using n-grams. ROUGE is developed for assessment of summarizing text quality and based on n-grams, furthermore compared to BLEU it is based on recall, so it assesses how generated tokens are relevant to reference ones. Since the required information may be contained in several documents and the recall of the required information is important, the ROUGE-L metric evaluating the longest common subsequence is used to assess the quality of the d_j retrieving.

As a large language model for solving the response A generation subtask, the Lite model (with a limit on the number of tokens L = 4096) of the GigaChat from Sber, which is an ensemble of RuGPT-3.5 and FRED-T5 1.7B finetune models, was used (Zmitrovich et al., 2023). As of March 2024, the FRED-T5 model ranks 3rd in the ranking of large language models for text processing in Russian (Shavrina et al., 2020), second only to the model from the Mistral AI project (by a fraction of a thousandth) and the human benchmark.

To test the quality of generating answer A to question Q for a document from set D, the BERTScore metric is used, based on the cosine measure of similarity of BERT embeddings, which allows taking into account the semantic proximity of the generated result and the standard (Zhang et al., 2020).

The university's corporate wiki system is used as a data source, which stores documents that represent reference information or local regulations.

Texts obtained from the corporate wiki system in HTML or PDF format 38 instructions and regulations for students are splitted into 52 chunks using the tokenizer of the Sentence BERT cointegrated/rubert-tiny2, the number of tokens satisfies the limitation of the feature extraction model K = 2048 tokens. The chunks are stored in a database controlled by PostgreSQL using the PGVector extension for storing embeddings. Synchronization with documents from the corporate wiki system is carried out once a day.

Using chatbots on VK and Telegram, questions from students were collected; for each question, experts selected a fragment of a document from the corporate wiki system. Also, using the GigaChat from Sber, questions were generated for documents, without using standard forms (applications, consent, etc.) or blocked by the service chunks due to being blacklisted. For each question-document pair from the test sample, the answer to the question on the document was manually written. The final distribution of questions and document chunks is shown in Table 1.

Table 1: Distribution of test and train datasets objects.

Dataset	Question source	Question count	Unique chunk count
Train	Chatbot users, GigaChat	355	42
Test	Chatbot users	67	27

Test and train datasets are published on the Hugging Face.

3 RESULTS

To retrieve a document chunk d_j from set D, containing 52 document fragments from a corporate wiki system, embeddings were built using the TF-IDF method, the Doc2vec algorithm, the pretrained Sentence BERT model and the GigaChat API. When using TF-IDF and Doc2vec, texts are lemmatized, cleared of stopwords and punctuation using the ru_core_news_sm model from the SpaCy library. The Doc2vec algorithm uses the following parameters: vector_size=150, window=5, min_count=1.

As a representative of Sentence BERT, the cointegrated/rubert-tiny2 model, pretrained on a Russian-language text corpus, was considered and fine-tuned. Multiple negatives ranking was selected as a loss function, intended for a corpus of pairs of nearest texts (batch_size=8, num_epochs=10).

To evaluate each method using the Accuracy and ROUGE-L metrics, the found document chunks were compared with the reference ones from the test sample (see Table 2).

Table 2: Metrics of document retrieving results using embeddings.

Method	Acc.	ROUGE LCS Recall	ROUGE LCS Precision	ROUGE LCS Fmean
TF-IDF	0.45	<u>0.63</u>	0.54	0.54
Doc2vec	0.06	0.19	0.19	0.16
Sentence RuBERT	0.37	0.49	0.51	0.48
API GigaChat	<u>0.54</u>	0.61	<u>0.65</u>	<u>0.62</u>
Fine-tuned Sentence RuBERT	**0.70**	**0.77**	**0.76**	**0.75**

Vectors obtained using the GigaChat API outperform the embeddings of TF-IDF, Doc2vec and Sentence RuBERT in most metrics. Fine-tuning of the Sentence RuBERT model from the text on the set of questions for corporate documents generated by LLM increased the accuracy of document retrieving to answer the question.

The fine-tuned model was uploaded on the Hugging Face and used in a microservice that implements the RAG. To develop the microservice, Python 3 frameworks were used: SQLAlchemy for ORM queries to vector storage, GigaChain for assembling prompt (see Table 3) and sending request to LLM, Aiohttp for processing HTTP requests.

Table 3: Prompt template for generating answers using language large model.

Prompt template	Действуй как Вопрошалыч — виртуальный помощник студента ТюмГУ. Используй следующий текст в тройных кавычках, чтобы кратко ответить на вопрос студента.

	Не придумывай и не изменяй ссылки, адреса и телефоны. Если ответа в тексте нет, напиши "ответ не найден". \"\"\" {context} \"\"\" Вопрос студента: {question}
Variables	context — the chunk for answer the question; question — the user question.

The microservice provides an API to generate a response to a user message. For example, if a user asks the question “Как выбрать электив?” (“How to choose an elective?” in Russian), the generated answer and a link to a retrieved document, which describes the process of choosing elective disciplines at the university, from the corporate wiki system will return:

```
# curl -X POST http://qa:80/qa/ -H 'Content-Type: application/json' -d '{"question":"Как выбрать электив?"}'

{"answer": "Рекомендую почитать отзывы на сервисе «Отзывус» по ссылке: https://electives.utmn.ru. Там же можно написать о своем опыте изучения элективных дисциплин.",
"confluence_url": "https://confluence.utmn.ru/pages/viewpage.action?pageId=86479065"}
```

If the LLM reports that there was no response, an exception will return. Questions and answers are saved to a database for further analysis. Testing the RAG pipeline is shown in Table 4.

Table 4: The examined approach test results.

Test type	Count of samples	BERTScore Precision	BERTScore Recall
All generated cases	67	0.61	0.64
Not including cases when answer is not found	50	0.81	0.86

The test results were influenced by the quality of document retrieving and the quality of response generation. Separately, cases were considered when the large language model was able to generate an answer in the proposed text, that is, retrieving a document to answer the question was successful.

4 DISCUSSION

The hypothesis, that fine-tuning of the sentence embedding model on the set of questions to corporate documents that has been generated by the LLM will increase the documents retrieving accuracy for answering questions compared to using embeddings from a LLM service, has been confirmed.

In terms of semantic similarity, the generation results have moderate overlap with human question responses, but still leave room for improvement in both document retrieval and answer generation.

The following study examines questions and documents only in text format in Russian related to regulations of the educational process at the university. Owing to computational resource limits, only using third party LLM services was possible.

Developed microservice is being used by chatbots in VK and Telegram of the UTMN Virtual Assistant that is being implemented at the University of Tyumen for replacing Student Affairs Office chatbot.

5 CONCLUSIONS

The Retrieval augmented generation has been examined for answering UTMN students’ questions by instructions and regulations published in corporate wiki system using natural language prompting for the large language model.

Students’ questions have been collected, the dataset of questions to corporate documents has been generated by LLM, feature extraction methods have been examined and Sentence RuBERT has been fine-tuned. The RAG microservice has been developed and tested.

Using named-entity recognition for increasing document extraction accuracy is planned to be in the future. In addition, transfer several document chunks to LLM input can be examined. At the same time, earlier sent questions and answers should also be taken into account in future work.

REFERENCES

Al Ghadban, Y., Lu, H. Y., Adavi, U., Sharma, A., Gara, S., Das, N., Hirst, J. E., 2023. Transforming healthcare education: Harnessing large language models for frontline health worker capacity building using retrieval-augmented generation. medRxiv, 2023-12.

Galeev, D., Panishchev, V. S., 2022. Experimental study of language models of "Transformer" in the problem of finding the answer to a question in a russian-language text. Informatics and Automation, 21(3), 521-542.

Hartrumpf, S., 2004. Question answering using sentence parsing and semantic network matching. In Workshop of the Cross-Language Evaluation Forum for European Languages (pp. 512-521). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ke, Y. H., Jin, L., Elangovan, K., Abdullah, H., Liu, N., Sia, A. T. H., Ting, D. S. W., 2024. Development and Testing of Retrieval Augmented Generation in Large Language Models.

Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196). PMLR.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kiela, D., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.

Lin, C. Y., 2004. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out (pp. 74-81).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26.

Papineni, K., Roukos, S., Ward, T., Zhu, W. J., 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics (pp. 311-318).

Reimers, N., Gurevych, I., 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. Shavrina, T., Fenogenova, A., Emelyanov, A., Shevelev, D., Artemova, E., Malykh, V., Evlampiev, A., 2020. RussianSuperGLUE: A Russian language understanding evaluation benchmark. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 4717–4726). Association for Computational Linguistics.

Sparck Jones, K., 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of documentation, 28(1), 11-21.

Sukhanova, N. T., Vezhelis, T. M., 2022. Using Chatbots to Automate the Provision of Reference Information to Applicants and University Students [Ispolzovanie chat-botov dlya avtoma-tizatsii predostavleniya spravochnoy informatsii abiturientam i studentam vuzov]. Problems of Modern Pedagogical Education [Problemy sovremennogo pedagogicheskogo obrazovaniya], 178-181.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

Wang, G., Liu, X., Ying, Z., Yang, G., Chen, Z., Liu, Z., Chen, Y., 2023. Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial. Nature Medicine, 29(10), 2633-2642. Zmitrovich, D., Abramov, A., Kalmykov, A., Tikhonova, M., Taktasheva, E., Astafurov, D., Fenogenova, A., 2023. A family of pretrained transformer language models for Russian.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., Artzi, Y., 2020. Bertscore: Evaluating text generation with bert. ICLR (2020).