2024 IEEE 18th International Conference on Application of Information and Communication Technologies (AICT) | 979-8-3503-8753-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/AICT61888.2024.10740455

Enhanced document retrieval with topic embeddings

Jafar Isbarov
Department of Computer Science
George Washington University
Washington, D.C., the U.S.
0000-0001-8404-2192

Kavsar Huseynova
Information Technology Department
Baku Higher Oil School
Baku, Azerbaijan
0009-0007-0362-9591

***Abstract*—Document retrieval systems have experienced a revitalized interest with the advent of retrieval-augmented generation (RAG). RAG architecture offers a lower hallucination rate than LLM-only applications. However, the accuracy of the retrieval mechanism is known to be a bottleneck in the efficiency of these applications. A particular case of subpar retrieval performance is observed in situations where multiple documents from several different but related topics are in the corpus. We have devised a new vectorization method that takes into account the topic information of the document. The paper introduces this new method for text vectorization and evaluates it in the context of RAG. Furthermore, we discuss the challenge of evaluating RAG systems, which pertains to the case at hand.**

***Index Terms*—Information retrieval, document retrieval, text embeddings**

I. Introduction

A. Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) systems allow us to create chatbots on large text corpora, such as law corpus, textbooks, and software documentation. RAG was introduced by [1]. It has seen a sudden rise in popularity due to the availability of improved large language models (LLMs), especially since the release of LLaMA models [2].

RAG system works as follows: (1) Text corpus is split into chunks and each chunk is vectorized. (2) User query is vectorized. (3) A similarity search is performed to find the chunk closest to the vectorized query. (4) The retrieved chunk is fed to the LLM along with the user query. (5) LLM uses this input to generate a free-form response.

One of the main bottlenecks in the performance of RAG systems is the retrieval step. The accuracy of the similarity search depends on multiple factors, including the choice of the similarity search algorithm, embedding method, and size of the indexed corpus. Corpus size becomes especially problematic if we are dealing with very similar documents.

B. Proposal

In most cases, we are not dealing with raw text data. Instead, the text comes with ample metadata that can be used to boost the performance of the similarity search. Our proposal is to use the available topic information of each chunk (i.e. document) and add it to the retrieval process. We have successfully implemented two such methods in industrial settings, and this paper attempts to generalize and evaluate these methods. The first method relies on creating a new document embedding by combining the original document embedding with the topic embeddings generated from the entire topic. The second method consists of two steps (1) find the topic, and (2) find the document within the topic.

Our contributions in this paper are as follows:

1) Propose two new methods for using topic metadata during the document retrieval process.
2) Evaluate the suggested methods with calculating the distance measures on embedded documents.
3) Propose a detailed problem statement for the next stages of this work.

The paper is structured as follows: The next section contains a review of relevant literature on these topics. The third section provides a detailed explanation of the proposed methods. The fourth section consists of two parts. First part outlines our experiments, their explanations, and results. A comprehensive discussion of our results, the main shortcomings of our work, as well as suggestions for future research take place in the second part. We conclude with the paper in the fifth section.

II. Related Work

RAG was introduced by [1] in 2020. Their work underpins more advanced RAG systems that we see today. [3] uses this idea to create a code summarization tool. Multimodal RAG systems are also seeing increasing popularity. [4] implements a competitive image and caption generation system with RAG architecture.

Vectorization is an integral part of RAG systems. Traditional vectorization techniques include bag-of-words [5], which simply uses word frequency and term frequency-inverse document frequency [6], which additionally takes into account how unique the word is to a particular document.

Relatively modern approaches include Word2Vec, GloVe, and fastText. Word2Vec [7] can use either continuous bag-of-words or continuous skipgram models. GloVe [8] is a log-bilinear regression model that was trained on a word-word cooccurrence matrix. fastText [9] is a skip-gram model that treats words as a combination of character n-grams.

The latest family of text vectorization methods relies on transformer-based language models. BERT [10] is a textbook example of this. These models usually use a masked attention mechanism to learn the context of tokens in a text. This information can later be used in various tasks, such as named entity recognition and sentiment analysis. The main advantage

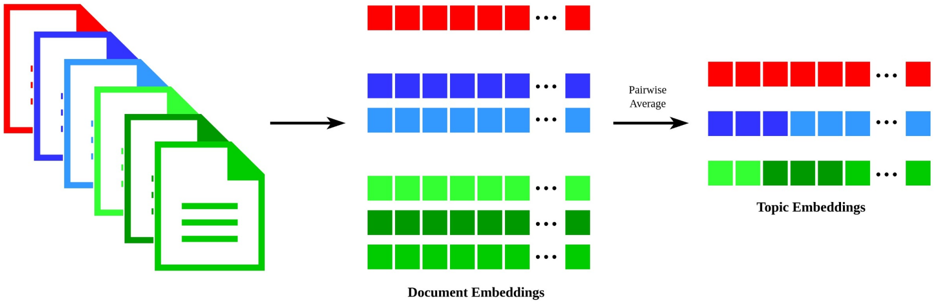


Fig. 1: Creating topic embeddings.

of this approach is that embeddings are contextual, i.e. they change depending on their place in the text. This is not the case with the aforementioned methods.

[11] exploits inherent hierarchical structure in the data during the embedding process. Our work is similar but distinct: We use explicit hierarchy, not implicit. As far as we know, no work has attempted this. [12] attempts a similar approach to enhance the performance of image classification models on ImageNet.

All of the mentioned works are either language-agnostic, or English-specific. Our dataset is in Azerbaijani, therefore works for the Azerbaijani language are of special interest to us.

Among the open-source embedding models, some claim to have an understanding of the Azerbaijani language. Google has released a multilingual version of the famous BERT model. We are aware of several unpublished attempts to use this model (directly or by fine-tuning) for various NLP tasks in Azerbaijani, all of them with limited success. Due to this, we avoided using the model as an embedder. Another multilingual model that has some understanding of Azerbaijani is mGPT. This is an unofficial version of the GPT-2 model released by [13] that was trained in a text corpus consisting of multiple languages. Its text generation capabilities in Azerbaijani are not satisfactory. This is why we did not use it in the generation part of our RAG systems.

III. Methods

We propose two new methods that use topic information during document retrieval: topic-enhanced document embeddings, and two-stage document retrieval.

A. Topic-enhanced document embeddings

Here, we create topic embeddings and then use these embeddings to update the original document embeddings in one of two ways.

Step 1: Create document embeddings.

This is part of the traditional RAG pipeline. We simply chunk the document and embed these chunks separately.

Step 2: Separate documents into topics.

We assume the topic information is provided either implicitly or explicitly. In any case, we need to have explicit topic labels by the end of this stage.

Step 3: Creating topic embeddings.

Topic embeddings are supposed to be a vector of the same size as document embeddings. If we are using a neural network like BERT to embed the text, we usually cannot feed the entire topic into the model. We can bypass this by taking an element-wise average of all document embeddings for that topic. If we are using a statistical method like TF-IDF, we can run it on the entire text regardless of the text length. However, we expect both the original document embeddings and the topic embeddings to use the same embedding method. Figure 1 visualizes the process of obtaining topic embeddings from documents.

Step 4: Update the document embeddings

We propose two alternate versions here:

1) *Average method*: Take an average of document embeddings and topic embeddings.
2) *Append method*: Concatenate document embeddings and topic embeddings.

The second method creates a new problem because now the embedding dimension of a query does not match the dimensions of our embedded documents. To solve this problem, we can duplicate the length of our query embeddings.

B. Two-stage document retrieval

This is a much simpler approach. We create topic embeddings just like the first method, but we perform retrieval in two stages:

1) Retrieve a topic based on the topic-embeddings.
2) Retrieve a document within that topic.

However, two-stage retrieval system comes with own challenges. One of which is increased inference time.

IV. Results

A. Experiments

To evaluate these approaches, we have curated a dataset from Azerbaijani laws. We have selected 14 major laws and

TABLE I: Azerbaijani laws in our dataset.

Law
Criminal Code
Code of Criminal Procedure
Customs Code
Constitution
Forest Code
Civil Procedure Code
Civil Code
Migration Code
Water Code
Land Code
Tax Code
Code on Administrative Violations
Labor Code
Education Law

split them into chunks of 2000 characters. You can find these laws in Table I. OpenAI's ”text-embedding-3-small” embedding model has been used to embed the chunks. Each law is treated as a distinct topic. We then used both the average method and the append method to update the document embeddings.

By treating topics (i.e., laws) as cluster labels, we have evaluated the ”clustering results” of three methods:

- original embeddings
- averaged embeddings
- appended embeddings

To visualize the embeddings of different methods, the tSNE algorithm has been used to reduce the number of dimensions to 2 [14]. You can see the results in Figure 2a and 2b.

We have calculated the performance of these ”clustering models” using three different metrics:

- Silhouette Coefficient
- Davies–Bouldin index
- Calinski–Harabasz index

The Silhouette Coefficient, ranging from -1 to +1, measures clustering effectiveness, with higher values indicating better clustering. In contrast, the Davies-Bouldin Index (DBI) assesses cluster separation and compactness, where lower values signify better quality. The Calinski-Harabasz (CH) Index evaluates how well-separated and dense clusters are, with higher values indicating better clusters, typically identified by a peak in the index. Together, these metrics provide a comprehensive assessment of clustering performance.

The results are available in Table II. As you can see, adding topic embeddings to the original embeddings results in better separation of different topics. The average method performs better than the append method, although we suspect that it may be data-specific. As future work, these experiments can be performed on larger and more variable datasets to assess the methods' performance against each other.

We were unable to evaluate the 'Two-stage document retrieval' method due to limitations in the evaluation dataset, which will be discussed further.

TABLE II: Performance of original and topic-based clustering. DBI: Davies–Bouldin index, CHI: Calinski–Harabasz index.

Metric	Original	Average	Append
Silhouette	0.01	**0.11**	0.06
DBI	4.60	**2.30**	3.25
CHI	63.42	**253.67**	126.84

B. Challenges and limitations

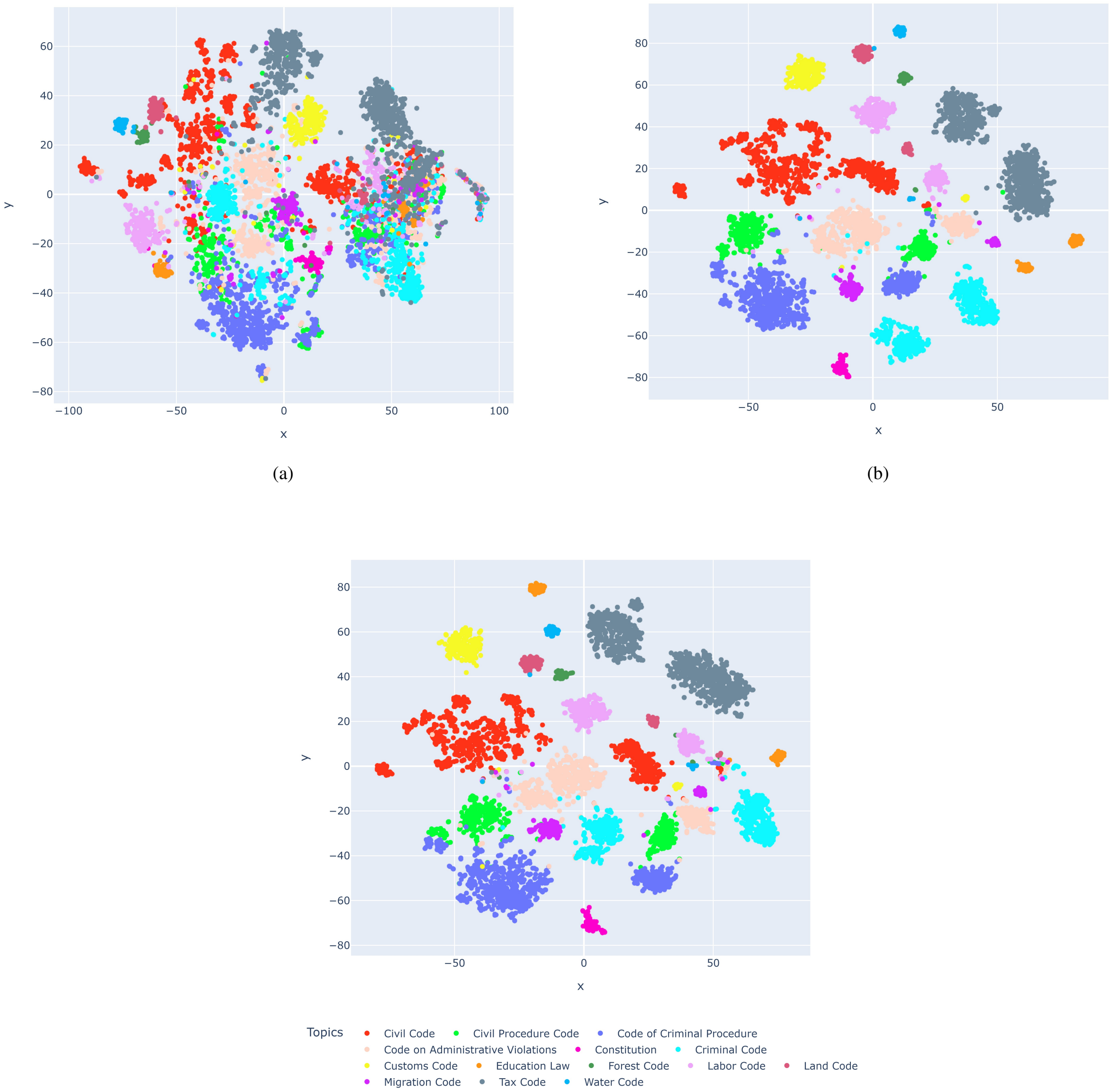
The main challenge we are facing is the lack of an end-to-end evaluation dataset. This dataset needs to have the following features:

- A text corpus with topic labels
- Natural queries that can be answered based on a part of this corpus.

Using this dataset, one can first embed the queries with traditional and our proposed methods, then run the similarity search on each of them to match the queries with respective documents, and hence get an evaluation score to make a fair comparison between traditional RAG system and our approach.

This first feature is easy to attain. We have already performed some analysis on such datasets. We have attempted to gain the second feature by synthetic query generation, but it did not achieve any better result than normal RAG system, because generated queries are highly specific to the chunk provided. For example, we used the following query:

> ***”Given the following context, generate a question that would be asked by a curious citizen:***
> *Context:*
> *Article 25. Right to equality*
> *I. Everyone shall be equal before the law and the courts.*
> *II. Men and women possess equal rights and freedoms.*
> *III. The State shall guarantee the equality of rights and freedoms to everyone, irrespective of race, ethnicity, religion, language, sex, origin, property status, occupation, beliefs or affiliation with political parties, trade union organisations or other public associations. Restrictions of rights and freedoms on the grounds of race, ethnicity, religion, language, sex, origin, beliefs, or political or social affiliation are prohibited.*
> *IV. No one may be harmed, granted advantages or privileges, or refused to be granted advantages and privileges on the grounds laid down in Paragraph III of the present Article.*
> *V. Everyone shall be guaranteed equal rights in any proceeding before state authorities and bearers of public authority that decide upon his/her rights and duties.*
> *VI. Persons with impaired health are entitled to all rights and carry all duties enshrined in this Constitution, except in cases when enjoyment of*



(a)

(b)

(c)

Fig. 2: 2D visualization of topics with (a) original embeddings, (b) averaged embeddings, (c) appended embeddings.

> *rights and performance of duties is impeded by their limited abilities."*

This resulted in the following response:

> *"What measures are in place to ensure enforcement of Article 25, particularly regarding equality before the law and in courts, as well as the guarantee of equal rights and freedoms regardless of various personal characteristics?"*

As you can see, if we generate a large dataset automatically using these prompts, we will end up with questions that exactly match the provided context (i.e., chunk). Because the content of the queries are highly related with the document chunks, our proposed methods achieve similar performance as the traditional RAG system.

Although the synthetic query generation attempts did not work, we have tried it on the English version of our dataset. As far as we are concerned, no LLM can generate natural text in Azerbaijani at a level that would be sufficient for this task. Due to these restrictions, we are researching the possibility of creating a natural dataset. We can use logs of one of our chatbots in production for this, or we can create an annotator team to build the dataset.

Another inherent limitation of our approach is that it depends on the existence of topic metadata. It would be interesting to devise a method to infer the topic information from the raw data itself, although we do not believe that there is a generalizable approach to this problem.

V. Conclusion

This paper introduces an enhanced vectorization technique for document retrieval. As RAG applications become increasingly popular, it is important to handle the challenge of having a large number of documents within the same database. We proposed two new methods that utilize the provided topic information in the text corpus. We have shown that by implementing our two novel approaches, the distance between different documents can be broadened through the introduction of topic embeddings, potentially resulting in more accurate similarity searches. Despite our progress in RAG application, there is a need to refine evaluation techniques. To ensure accurate and representative performance assessments of our research, an end-to-end evaluation dataset should be curated.

Acknowledgements

This work has emerged out of projects developed in PRODATA LLC. It has later been developed in aLLMA Lab.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[2] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023.

[3] S. Liu, Y. Chen, X. Xie, J. Siow, and Y. Liu, "Retrieval-augmented generation for code summarization via hybrid gnn," 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:234487049

[4] M. Yasunaga, A. Aghajanyan, W. Shi, R. James, J. Leskovec, P. Liang, M. Lewis, L. Zettlemoyer, and W.-t. Yih, "Retrieval-augmented multimodal language modeling," ser. ICML'23. JMLR.org, 2023.

[5] W. A. Qader, M. M. Ameen, and B. I. Ahmed, "An overview of bag of words;importance, implementation, applications, and challenges," in *2019 International Engineering Conference (IEC)*, 2019, pp. 200–204.

[6] S. Qaiser and R. Ali, "Text mining: Use of tf-idf to examine the relevance of words to documents," *International Journal of Computer Applications*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:53702508

[7] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *International Conference on Learning Representations*, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:5959482

[8] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162

[9] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:207556454

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:52967399

[11] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[12] B. Barz and J. Denzler, "Hierarchy-based image embeddings for semantic image retrieval," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 638–647.

[13] O. Shliazhko, A. Fenogenova, M. Tikhonova, V. Mikhailov, A. Kozlova, and T. Shavrina, "mgpt: Few-shot learners go multilingual," 2023.

[14] T. T. Cai and R. Ma, "Theoretical foundations of t-sne for visualizing high-dimensional clustered data," *J. Mach. Learn. Res.*, vol. 23, no. 1, jan 2022.