Semantic Embeddings for Arabic Retrieval Augmented Generation (ARAG)

Hazem Abdelazim
School of Computing and
Digital Technology
ESLSCA University
Cairo, EGYPT

Mohamed Tharwat
School of Computing and
Digital Technology
ESLSCA University
Cairo, EGYPT

Ammar Mohamed
School of Computing and
Digital Technology
ESLSCA University
Cairo, EGYPT

***Abstract*—In recent times, Retrieval Augmented Generation (RAG) models have garnered considerable attention, primarily due to the impressive capabilities exhibited by Large Language Models (LLMs). Nevertheless, the Arabic language, despite its significance and widespread use, has received relatively less research emphasis in this field. A critical element within RAG systems is the Information Retrieval component, and at its core lies the vector embedding process commonly referred to as "semantic embedding". This study encompasses an array of multilingual semantic embedding models, intending to enhance the model's ability to comprehend and generate Arabic text effectively. We conducted an extensive evaluation of the performance of ten cutting-edge Multilingual Semantic embedding models, employing a publicly available ARCD dataset as a benchmark and assessing their performance using the average Recall@k metric. The results showed that the Microsoft E5 sentence embedding model outperformed all other models on the ARCD dataset, with Recall@10 exceeding 90%**

Keywords*—Arabic NLP; large language models; retrieval augmented generation; semantic embedding

I. Introduction

Retrieval Augmented Generation (RAG), introduced by Facebook Researchers in 2020 [1], is a pivotal AI framework facilitating information retrieval for Generative AI models, thereby enhancing their accuracy and capabilities. RAG empowers Large Language Models (LLMs) by granting them access to external knowledge sources, augmenting the content generation process. This dual functionality entails retrieval, wherein RAG meticulously selects pertinent information from provided sources and generation, whereby LLMs craft contextually relevant responses based on user input.

The advantages of RAG are multi-fold. Firstly, it bolsters the performance by grounding LLMs with factual, up-to-date information from external knowledge repositories. Furthermore, RAG maintains contextual relevance in responses, contributing to a more engaging user experience in conversational AI applications. Its scalability is noteworthy, as RAG models seamlessly handle copious volumes of information, proving invaluable for data-intensive tasks. Additionally, the adaptability of RAG models allows fine-tuning for specific applications [2], rendering them versatile across diverse data and use cases. Customizability is another hallmark, permitting RAG models to specialize in particular domains or subjects through customization and fine-tuning on specific knowledge bases. Due to the importance of such a framework for enterprises, extensive research is currently being pursued to discover new algorithms and techniques to enhance the performance of such models bounded by the context-window limitations of LLMs. Although there is ongoing research to expand the window size for LLM to be able to ingest more data in the prompt, the use of techniques like RAG is still of great practical importance, not only on homogeneous unstructured data but also on heterogeneous data [3].

In principle, at the heart of the information retrieval module is the semantic embedding module which converts a piece of text, whether a query or a context text chunk to a numeric feature vector that embodies all semantic features of the text. The development of word and sentence embeddings is a relatively recent area of research in natural language processing (NLP) and information retrieval.

Most of the semantic models are English language-centred; however, in recent years, Multilingual embedding models were released [4]. There are lots of benchmarks to test the performance of multilingual embeddings [5], which are aggregate but very few focus on language-specific performance, and on the Arabic language in particular. This is the main impetus behind the current research work, which focuses on ten different state-of-the-art embedding models that are capable of embedding Arabic language.All the models are tested using publicly available ARCD (Arabic Reading Comprehension Dataset) [6] and the metric used is average Recall@k for different values of k. A comparative performance is conducted taking into consideration the embedding size for each model.

The rest of the paper is organized as follows: Section II, the Retrieval Augmented generation pipeline is presented, as well as the positioning of the semantic embedding and the information retrieval component within the pipeline. Section III explores related research work in this field, focusing on recent developments. Section IV overviews the 10 semantic embedding models that are used in the experiments will be covered. Section V discusses the 10 embedding models on a standard dataset that are used in for Arabic Reading comprehension (ARCD) and their evaluation using Recall@k performance metric. Also, the impact of the embedding dimension size is analyzed in the comparative results. Section VI concludes the paper.

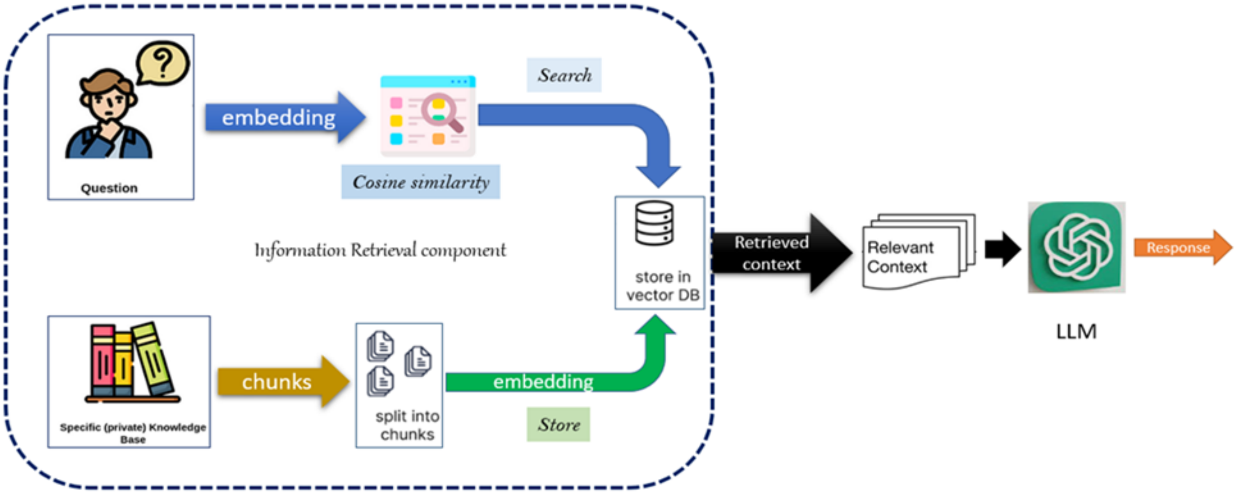


Fig. 1. Retrieval augmented generation.

II. RAG: Retrieval Augmented Generation

The Retrieval Augmented generation pipeline, shown in Fig. 1 is as follows:

A. Phase I: Information retrieval

1) Given a corpus of unstructured text containing documents
2) Given a user text query
3) A semantic embedding model is identified
4) The query is embedded into a feature vector of n dimension (semantic embedding)
5) The corpus is segmented into m text chunks (either disjoint or overlapping)
6) Each text chunk is embedded into a feature vector of n dimension using the same embedding model used in embedding the search query, as shown in Fig. 2.
7) The m- m-vectors are indexed in a Vector DB store
8) Cosine similarity, euclidean or inner product score is computed between the embedded vector of the query
9) 9. Top k relevant chunks are retrieved, which comprise a context for the next phase

B. Phase 2: LLM Comprehension and Response

In this phase, a suitable LLM is identified and selected, whether an open source model (more than 100 LLM models are currently available) , like LLaMA (7b/13b/70b), Falcon, GPT neoX, Bloom, vicuna , FlanT5 , etc.) . However, not all of them support the Arabic Language. The Current Arabic LLM models are:

- OpenAI GPT-turbo-3.5
- Open AI GPT 4.0
- Google Bard
- Microsoft Bing Chat (on top of openAIGPT3.5)
- Google PaLM2 (vertex-ai)
- Jais (UAE Arabic Language Model)

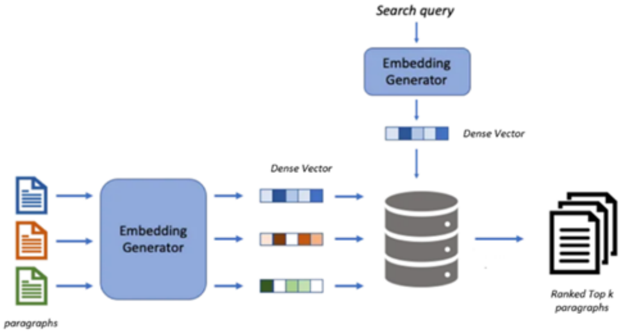


Fig. 2. Semantic embedding.

However, not all of them provide APIs for programmatic tasks, which are provided at cost like (OpenAI GPT3.5-turbo/GPT4.0 Google PalM - vertex-ai). After an Arabic LLM is identified. A prompt is constructed with two variable components: the retrieved Top k text chunks as a context and the input research query. The prompt instructs the Arabic LLM to find an answer to the search query from within the retrieved context. This architecture is widely used in Enterprises for domain-specific deployment of generative AI LLMs. A fundamental component of the overall process is the information retrieval component, as depicted in Fig. 1. The overall efficiency of the system is highly dependent on the performance of the IR component. If the IR component fails to retrieve relevant portions from the corpus, the LLM will not find the proper answer or, even worse, may hallucinate with wrong answers confidently depending on the LLM model settings (like temperature) as well as the prompt engineering. Ensuring an efficient and accurate semantic embedding is of paramount importance for an effective and practical RAG system.

III. Related Work

In the context of Arabic language processing, a lot of research was done in Natural Language Understanding, taking into consideration the different dialectical nature of Arabic language [7], in addition to research work on Arabic text classification [8] as well as Arabic text similarity using statistical techniques [9]. However, relatively fewer research studies were conducted on semantic embeddings for Arabic text. FastText for Arabic Word Embeddings [11] proved to be an effective method for generating Arabic word embeddings. These embeddings capture subword information, making them valuable for morphologically rich languages like Arabic. Researchers have also explored Word2Vec-based approaches for Arabic word embeddings [12].

Multilingual FastText: Multilingual embeddings have gained attention for their ability to handle multiple languages simultaneously through incorporating language-specific information while sharing a common subword vocabulary across languages [4]. Researchers have also explored cross-lingual embeddings that facilitate knowledge transfer between languages. A new method is proposed that aligns word embeddings across languages, enabling multilingual applications [13].

Sentence-BERT: Sentence embeddings, which capture the semantic meaning of entire sentences, have gained popularity [10] and demonstrated superior performance in various sentence-level tasks in monolingual settings. Multilingual sentence embeddings [14] have been explored for cross-lingual applications based on training sentence embeddings for multiple languages using a shared model. The research work in [9] provided a foundational framework for evaluating critical semantic embedding APIs that play a pivotal role in search and broader information access initiatives. The author in [9] addresses the challenge of the limited accessibility of increasingly large language models by examining the utilization of semantic embedding APIs for information retrieval. Their investigation focused on assessing the capabilities of these APIs in domain generalization and multilingual retrieval using benchmark datasets like BEIR and MIRACL. The study reveals that re-ranking BM25 results using these APIs proves to be cost-effective and most effective in English contexts, offering an alternative to the conventional practice of using them as initial retrievers. For non-English retrieval, the authors suggest a hybrid model with BM25 as the most effective approach, albeit at a higher cost.

For using embeddings in downstream tasks, the authors in [17] focused on Arabic sentiment analysis, particularly on social media platforms like Twitter and Facebook, which have become vital for understanding user opinions and preferences. Sentiment analysis, however, faces challenges in natural language processing (NLP). Recent advancements in deep learning have demonstrated superior performance in NLP-related tasks compared to traditional statistical and lexical-based approaches. A comparative analysis of classic and contextualized word embeddings for sentiment analysis was conducted utilizing both trained and pre-trained versions of the four most commonly used word embedding techniques: GloVe, Word2Vec, FastText, and ARBERT. Deep learning architectures, namely, BiLSTM and CNN, are employed for sentiment classification, and experiments are conducted on benchmark datasets, including HARD, Khooli, AJGT, ArSAS, and ASTD. The results reveal that, in general, embeddings generated by one technique outperform their pre-trained counterparts, with contextualized transformer-based embedding BERT achieving the highest performance, highlighting the significance of word embeddings in Arabic sentiment analysis.

An Arabic reading comprehension dataset (ARCD) [6] addressed the challenge of open-domain Arabic question and answering (QA) with Wikipedia as the knowledge source. Mainly the scarcity of labelled QA datasets and the need for efficient Arabic machine reading comprehension and retrieval. To overcome the lack of Arabic QA datasets, they introduced the Arabic Reading Comprehension Dataset (ARCD), generated by crowd-workers from Wikipedia articles and a machine translation of the Stanford Question Answering Dataset (Arabic-SQuAD). Their open-domain QA system, SOQAL, included two components; the first is a hierarchical TF-IDF component and a neural reading comprehension component based on the pre-trained BERT transformer. Experiments on ARCD demonstrate the effectiveness of their approach, with the BERT-based reader achieving a 61.3 F1 score and SOQAL achieving a 27.6 F1 score in open-domain Arabic question answering.

In the next session, we will dive more into the semantic embedding models used in the current research.

IV. Semantic Embedding Models

Sentence and paragraph embeddings are crucial tools in information retrieval (IR), enabling systems to comprehend and retrieve text based on semantic meaning. These embeddings encode the meaning of sentences and paragraphs into fixed-size vectors [16], [18], allowing for semantic search, document retrieval, question answering, duplicate detection, clustering, summarization, recommender systems, cross-lingual search, and contextual understanding. By representing queries and documents in a continuous vector space, these embeddings enhance the accuracy of IR tasks by retrieving relevant content, even when keyword matching falls short in capturing the nuances of user intent or dealing with extensive and unstructured text collections. In the Question and Answering (QA) setting under study: Given a complete dataset of records (context-paragraphs (cps), question, ground truth answer), the IR problem is to retrieve the most relevant cps to this query. In the current work, since our focus is on the Arabic language, we explored ten embedding models that have multilingual embedding features. The query is embedded, resulting in a fixed-size feature vector, and each of the context paragraphs (cps) is also embedded. The result of embedding is a feature vector that embodies the semantic features of the text (question or context paragraph). A cosine similarity distance metric is calculated between the query, and all the semantic features of all the cps is given by

$$\text{cosine\_similarity}\left(\mathbf{A},\mathbf{B}\right)=\frac{\sum_{i=1}^{n}A_i\cdot B_i}{\sqrt{\sum_{i=1}^{n}A_i^2}\cdot\sqrt{\sum_{i=1}^{n}B_i^2}} \quad (1)$$

The cosine similarity metric is a value between [0,1]; the higher score implies a higher similarity. The essence here is that “most probably the answer of the query lies in the ‘context-paragraph’ cp with the highest similarity with the input query”. This assumption, which is mostly adopted in current QA systems, works well in the majority of situations with much higher performance than keyword search and retrieval. The following multilingual embedding models were investigated in this research work.

A. Mpnet: Paraphrase-Multilingual-mpnet-base-v2

Mpnet [10] is based on SBERT (Sentence-BERT), which is a modification of the pre-trained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings, allowing for efficient comparisons using cosine similarity. The original work BERT [18] and RoBERTa [19] have achieved state-of-the-art performance on sentence-pair regression tasks such as semantic textual similarity (STS). However, they require both sentences to be fed into the network, leading to significant computational overhead. Siamese networks are a type of neural network architecture that can learn to compare two inputs and measure their similarity or dissimilarity. BERT is a pre-trained language model that can encode sentences into fixed-length vectors, but it requires both sentences to be fed into the network simultaneously, which is inefficient for large-scale applications. Sentence-BERT (SBERT) is a modification of BERT that uses Siamese networks to derive sentence embeddings that can be compared using cosine similarity. This way, SBERT can compute the similarity of two sentences without processing them together, reducing the computational cost and enabling semantic similarity search and clustering. SBERT can produce more accurate and consistent embeddings than BERT, as it fine-tunes the model on specific similarity tasks.

B. Google LaBSE

While BERT has proven to be a powerful approach for acquiring monolingual sentence embeddings that excel in tasks related to semantic similarity and embedding-based transfer learning, the realm of BERT-based cross-lingual sentence embeddings was relatively uncharted. A comprehensive Language-agnostic BERT Sentence Embedding (LaBSE) [20], developed by Google researchers with training across 112 languages, including Arabic, using the Tatoeba dataset [13]. This is the multilingual SBERT model used in this research work. The embedding dimension is 768.

C. Openai Ada-embedding

Openai research team [21] delved into the significance of text embeddings, essential for tasks such as semantic search and text similarity assessment, transcending traditional applications. Unlike previous methods that tailored models for specific use cases, they introduced a more unified approach, emphasizing extensive contrastive pre-training on unsupervised data. This strategy produced top-quality vector representations with a wider context window for both text and code, a breakthrough validated across various benchmarks, including MSMARCO [15], Natural Questions, TriviaQA, and code search. Their findings underscored the versatility of these unsupervised text embeddings, demonstrating their potential to excel in state-of-the-art performance across diverse domains, from linear-probe classification to large-scale semantic search.

D. Cohere Multilingual Embedding

Cohere's multilingual text understanding model [16] works by mapping text to a semantic vector space, where texts with similar meanings are positioned close to each other. This allows for a variety of valuable use cases in multilingual settings, such as search, content aggregation and recommendation, and zero-shot cross-lingual text classification. To train the model, Cohere collected a dataset of nearly 1.4 billion question/answer pairs across tens of thousands of websites in hundreds of languages. This dataset is unique because it contains questions actually asked by speakers of said languages, allowing the model to capture language- and country-specific nuances.

E. Meta SONAR:Language-Agnostic Representations

Meta introduced SONAR, a novel fixed-size sentence embedding space with support for multiple languages and modalities [22]. SONAR's single text encoder, spanning 200 languages. Meta stipulated that SONAR outperforms existing sentence embeddings like LASER3 and LabSE in multilingual similarity search tasks. It extends its capabilities to speech segments by employing language-specific speech encoders trained in a teacher-student framework, surpassing existing speech encoders in similarity search tasks. SONAR also provides a text decoder for 200 languages, facilitating text-to-text and speech-to-text machine translation, including zero-shot language and modality combinations [22]. In our findings, we found that this is an overstatement when applied to Arabic Language, as described in section 4.

F. Microsoft E5 - (Small-base-large)

Microsoft researchers [23] presented E5, a family of advanced text embeddings designed for versatile applications across various tasks. E5 stands for EmbEddings from bidirectional Encoder representations. These embeddings are trained using a contrastive approach applied to a large, curated text pair dataset called CCPairs. E5 text embedding models are suitable for tasks like retrieval, clustering, and classification, where a single-vector representation of text is required. It exhibits robust performance in both zero-shot and fine-tuned settings. The authors extensively evaluated 56 datasets using BEIR and MTEB [5] benchmarks. In zero-shot scenarios, E5 surpasses the strong BM25 baseline in the BEIR retrieval benchmark, and when fine-tuned, it achieved the best results in the MTEB benchmark at the time of the publication (Dec. 7th. 2022), outperforming existing embedding models with significantly fewer parameters.

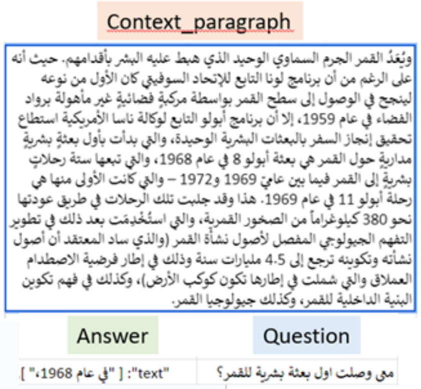
G. HuggingFace DistillBert v1,v2

HuggingFace researchers [24], proposed DistilBERT, a smaller and more efficient language representation model derived from BERT. As transfer learning from large-scale pre-trained models gains prominence in Natural Language Processing (NLP), the challenge lies in deploying these large models on resource-constrained devices or under tight computational budgets. DistilBERT is pre-trained using knowledge distillation techniques, reducing the model size by 40% while retaining 97% of its language understanding capabilities and achieving a 60% increase in speed. To leverage the inductive biases from larger models, they introduced a triple loss mechanism that combines language modelling, distillation, and cosine-distance losses. DistilBERT proves to be cost-effective for pre-training and demonstrates its suitability for on-device computations through proof-of-concept experiments and comparative on-device studies. In our analysis, we explored distils-base-multilingual-cased-v1 and v2 , denoted in the experiments as hf1 and hf2

V. Experimental Results

A. ARCD Dataset

The dataset used in the benchmark analysis is ARCD (Arabic Reading Comprehension Dataset) [6]. Crowdsourced 1,395 questions with the corresponding context paragraph and ground truth answers. The research involved curation and crowdsourcing, focusing on 155 randomly selected articles from the top 1000 most viewed articles on Arabic Wikipedia in 2018. These articles spanned a wide range of topics, including religious figures, historical figures, sports celebrities, countries, and companies. To ensure the appropriateness of the content, a manual filter was applied to remove any adult material. In total, the project collected 1,395 questions based on 465 paragraphs extracted from the 155 selected articles. Fig. 3 shows a typical record from the ARCD dataset. Following the pipeline in Fig. 1, the knowledge base (corpus) is constructed based on the concatenation of all Context paragraphs (CPs) of all 1395 questions. Each question and each context paragraph CP is embedded using one of the ten models under study. Faiss (Facebook AI Similarity Search) python library is used for Vector DB indexing and search.



Context_paragraph

ويُعَدُّ القمر الجرم السماوي الوحيد الذي هبط عليه البشر بأقدامهم. حيث أنه
على الرغم من أن برنامج لونا التابع للإتحاد السوفيتي كان الأول من نوعه
لينجح في الوصول إلى سطح القمر بواسطة مركبةٍ فضائيةٍ غير مأهولة برواد
الفضاء في عام 1959، إلا أن برنامج أبولو التابع لوكالة ناسا الأمريكية استطاع
تحقيق إنجاز السفر بالبعثات البشرية الوحيدة، والتي بدأت بأول بعثةٍ بشريةٍ
مداريةٍ حول القمر هي بعثة أبولو 8 في عام 1968، والتي تبعها ستة رحلاتٍ
بشريةٍ إلى القمر فيما بين عامَي 1969 و1972 – والتي كانت الأولى منها هي
رحلة أبولو 11 في عام 1969. هذا وقد جلبت تلك الرحلات في طريق عودتها
نحو 380 كيلوغراماً من الصخور القمرية، والتي استُخدِمَت بعد ذلك في تطوير
التفهم الجيولوجي المفصل لأصول نشأة القمر (والذي ساد المعتقد أن أصول
نشأته وتكوينه ترجع إلى 4.5 مليارات سنة وذلك في إطار فرضية الاصطدام
العملاق والتي شملت في إطارها تكون كوكب الأرض)، وكذلك في فهم تكوين
البنية الداخلية للقمر، وكذلك جيولوجيا القمر.

Answer	Question
"text": ["في عام 1968،"]	متى وصلت اول بعثة بشرية للقمر؟

Fig. 3. ARCD record example.

B. Recall@k Performance Metric

Average Recall@k metric is used, where k is the top model retrieval hits with values [1,2,3,4,5,10,15]. The performance results are shown in Fig. 4 and Table I.

The results showed that the Microsoft E5 Family of models had a superior overall performance for all k values, where the top performer was E5 - ML - Large. the second was ada openAI embedding, and worth noting here that E5 is a free, open-source model, while Openai ada is at a cost (0.0001$/1k tokens)

TABLE I. Recall@k for Various Models

	sonar	e5s	e5b	e5l	hf1	hf2	cohere	mpnet	ada	LaBSE
$k = 1$	11%	21%	21%	23%	15%	14%	15%	14%	18%	15%
$k = 2$	23%	43%	42%	46%	27%	26%	28%	27%	36%	29%
$k = 3$	35%	63%	62%	68%	40%	38%	40%	38%	53%	44%
$k = 4$	40%	71%	69%	75%	46%	44%	44%	44%	59%	50%
$k = 5$	44%	76%	74%	79%	52%	49%	48%	49%	64%	55%
$k = 10$	56%	87%	87%	91%	66%	63%	60%	61%	77%	70%
$k = 15$	63%	90%	90%	93%	71%	67%	64%	65%	79%	78%
Embedding	1024	384	768	1024	512	512	768	768	1536	768

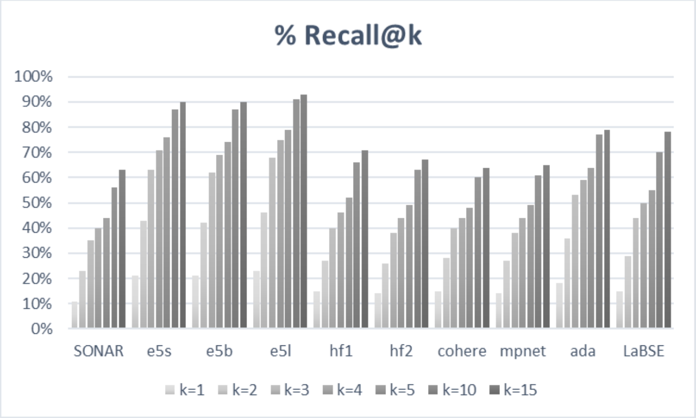


Fig. 4. Average % Recall@k performance.

C. Embedding Dimensions and Model Score

Fig 5 shows the embedding dimension of each model and the top model had 1024, while Openai ada had the maximum embedding dimension of 1536. What's interesting is that the second top performer, E5-small, has an embedding dimension of 384, which is quite impressive. Naturally, the higher the embedding dimension, the higher the capacity to capture better semantic context, which impacts storage and Latency. A simple formula is used to capture the trade-off between model retrieval accuracy and embedding dimension in an overall score:

$$model\_score = \frac{Avg.\ Recall@k\ \ * 1000}{Embedding\_Dimension} \tag{2}$$

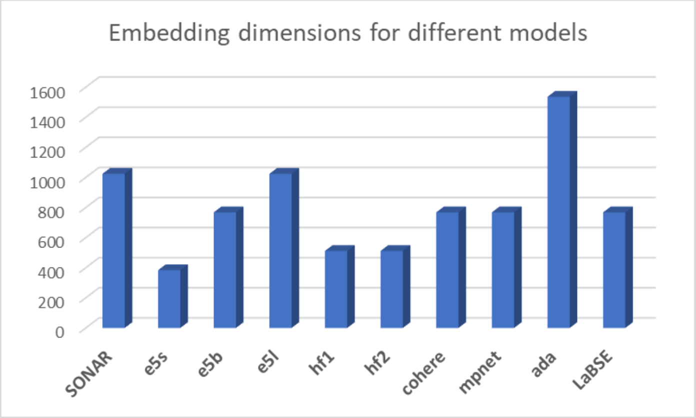


Fig. 5. Embedding dimensions.

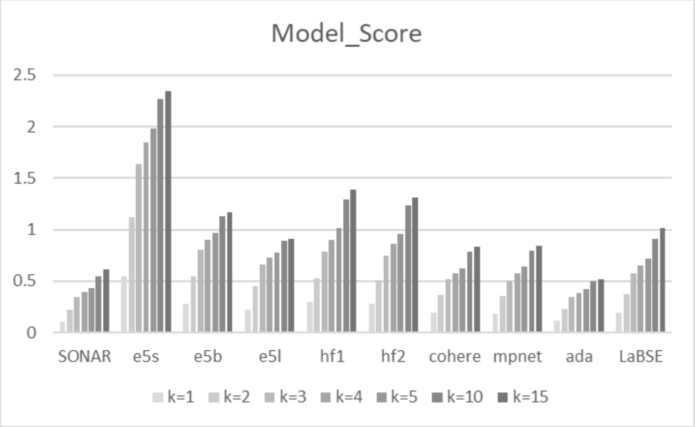


Fig. 6. Overall model_score.

The overall model_score in Fig. 6 shows a clear superior performance for the E5-ml-small model with 384 embedding dimensions, and hence highly recommended for Arabic Language Semantic Information retrieval

VI. Conclusion

In this study, we explored the application of Retrieval Augmented Generation (RAG) models in the realm of the Arabic language, an area that while linguistically rich, often receives less attention in this field. Our focus was particularly on the Information Retrieval component, with a keen eye on the processes of semantic embedding. For our evaluation, we utilized a range of advanced Multilingual Semantic embedding models, employing the ARCD dataset as a benchmark for our assessments. The knowledge corpus is generated from the concatenations of ARCD question contexts. Questions and contexts are embedded using the 10 models understudy, and Recall@k metric is used in the evaluation, where k represents the top retrieval hits based on the cosine similarity distance, and facebook AI similarity search (faiss) library.

The results indicated that the Microsoft E5 Family of models, especially E5-ML-Large (e5l), consistently outperformed the other models across different retrieval hit levels (k values). Notably, the open-source nature of E5 models makes them particularly appealing for a wide range of applications. The second-best performer was the Ada OpenAI embedding, albeit for 0.0001$/1k tokens. Furthermore, we observed that embedding dimensions play a crucial role in model performance. Higher embedding dimensions, such as the 1536 of OpenAI Ada, offer improved semantic context capture but come with storage and latency implications. To account for this tradeoff, we introduced an overall model score that combines model retrieval accuracy and embedding dimension. The E5-ML-Small model (e5s) , with an embedding dimension of 384, emerged as the top performer in this balanced evaluation. In light of these findings, we highly recommend the adoption of the E5-ML-Small model for Arabic Language Semantic Information Retrieval, as it strikes an excellent balance between retrieval accuracy and resource efficiency.

The superior performance of the e5 family of models is attributed to their unique approach to data preparation and training. Unlike conventional methods that rely on small-scale, human-annotated data or large-scale, noisy datasets, the e5 models utilize a specially curated dataset called CCPairs (Colossal Clean text Pairs), which is derived from diverse semi-structured sources.

This research contributes to the broader exploration of RAG models for Arabic language processing and information retrieval, shedding light on valuable avenues for future application of Arabic Language Understanding and Generation.

VII. Discussions and Future Work

A critical aspect warranting further investigation in Retrieval Augmented Generation (RAG) systems and semantic embeddings pertains to the dimensionality of the context window, or embedding size. While reduced embedding dimensions are advantageous for computational efficiency and data storage, they pose challenges in terms of model performance, particularly when processing extensive contexts. Such contexts often exceed the embedding dimension limits, leading to truncation which may adversely impact the model's effectiveness.

Moreover, the choice of tokenizer algorithm, inherently linked to the language being analyzed, presents another variable influencing RAG systems' performance. Tokenizer algorithms vary significantly, and their compatibility and efficiency can differ across languages. This variability underscores the necessity for extensive research into the implications of different tokenizer algorithms, especially in the context of specific languages. Such an investigation could provide valuable insights into optimizing RAG systems for diverse linguistic environments.

Future research should also encompass the exploration of contemporary methodologies in the realm of RAG systems, notably re-ranking strategies and cross-encoder architectures, from a language-specific perspective. This exploration is essential given the evolving nature of large language models and their application across various downstream tasks. In conducting such studies, it will be critical to employ nuanced, language-sensitive metrics, such as Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and normalized Discounted Cumulative Gain at k (ndcg@k). These metrics would offer a more refined evaluation of the models' capabilities in handling language-specific nuances and complexities.

Through these focused areas, we are aim to address the interplay between linguistic characteristics and the technical dimensions of RAG systems, thereby enhancing their applicability and efficiency in diverse linguistic contexts.

References

[1] Lewis, Patrick & Perez, Ethan & Piktus, Aleksandara & Petroni, Fabio & Karpukhin, et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". Advances in neural information processing in systems (2020)

[2] Krishna CS. Prompt Generate Train (PGT): A framework for few-shot domain adaptation, alignment, and uncertainty calibration of a retriever augmented generation (RAG) model for domain specific open book question-answering. arXiv preprint arXiv:2307.05915. 2023 Jul 12.

[3] Yu, W.. "Retrieval-augmented Generation across Heterogeneous Knowledge. North American Chapter of the Association for Computational Linguistics (2022).

[4] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

[5] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

[6] Mozannar, H., El Hajal, K., Maamary, E., & Hajj, H. (2019). Neural Arabic Question Answering. Proceedings of the Fourth Arabic Natural Language Processing Workshop, 108–118. Florence, Italy, August 1, 2019. © 2019 Association for Computational Linguistics.

[7] Muhammad Khalifa, Hesham Hassan and Aly Fahmy, “Zero-resource Multi-dialectal Arabic Natural Language Understanding” International Journal of Advanced Computer Science and Applications(IJACSA), 12(3), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0120369

[8] Alroobaea R. An Empirical Deep Learning Approach for Arabic News Classification. International Journal of Advanced Computer Science and Applications. 2023;14(6).

[9] Al-Mahmoud RH, Sharieh A. NGram Approach for Semantic Similarity on Arabic Short Text. International Journal of Advanced Computer Science and Applications. 2022;13(11).

[10] Reimers, Nils, and Iryna Gurevych. ”Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.” Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. November 2019.

[11] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

[12] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. ”Distributed representations of words and phrases and their compositionality.” Advances in neural information processing in systems, pp. 3111-3119. 2013.

[13] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

[14] S. Gouws and A. Søgaard. 2015. Simple task-specific bilingual word embeddings. In NAACL-HLT., pages 1386–1390

[15] Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L. (2016). ”MS MARCO: A Human Generated MAchine Reading COmprehension Dataset.” Retrieved November 2016, from https://www.microsoft.com/en-us/research/publication/ms-marco-human-generated-machine-reading-comprehension-dataset/

[16] Ehsan Kamalloo, Xinyu Zhang, Odunayo Ogundepo, Nandan Thakur, David Alfonso-hermelo, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. Evaluating Embedding APIs for Information Retrieval. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track), pages 518–526, Toronto, Canada. Association for Computational Linguistics.

[17] Sabbeh, Sahar & Fasihuddin, Heba. (2023). A Comparative Analysis of Word Embedding and Deep Learning for Arabic Sentiment Classification. Electronics. 12. 1425. 10.3390/electronics12061425

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. ”RoBERTa: A Robustly Optimized BERT Pretraining Approach.” CoRR, vol. abs/1907.11692.

[20] Mohammed Alsuhaibani. Deep Learning-based Sentence Embeddings using BERT for Textual Entailment. International Journal of Advanced Computer Science and Applications, 14(8), 2023. doi: 10.14569/IJACSA.2023.01408108. URL: http://dx.doi.org/10.14569/IJACSA.2023.01408108

[21] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, Lilian Weng. ”Text and Code Embeddings by Contrastive Pre-Training.” 2022. [Link](https://arxiv.org/abs/2201.10005)

[22] Duquenne, Paul-Ambroise (MetaAI & Inria), Schwenk, Holger (MetaAI), Sagot, Benoît (Inria). ”SONAR: Sentence-Level Multimodal and Language-Agnostic Representations.”, August 2023. Meta publisher

[23] Wang, Liang, Nan Yang, Xiaolong Huang,etal , microsoft., ”Text Embeddings by Weakly-Supervised Contrastive Pre-training.” 2022

[24] Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR abs/1910.01108 (2019).