Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 244 (2024) 66–75

Procedia
Computer Science

www.elsevier.com/locate/procedia

6th International Conference on AI in Computational Linguistics

Multi-Hop Arabic LLM Reasoning in Complex QA

Hazem Abdelazim[a], Tony Begemy[a], Ahmed Galal[a], Hala Sedki[a], Ali Mohamed[a]

[a]*School of Computing and Digital Technologies ESLSCA University Giza, Egypt, 11841*

Abstract

The introduction of Large Language Models (LLMs), and generative AI has significantly transformed the field of natural language processing. These models have exhibited profound reasoning capabilities, marking considerable progress across diverse general knowledge reasoning tasks. Consequently, the deployment of LLMs in domain-specific contexts has become a prime objective for governments and corporations eager to leverage the generative AI revolution. However, the Arabic language has notably lagged in attention and development compared to other languages in this arena. This research endeavors to delve into various facets of Arabic closed-domain question and answering systems that emulate the reasoning requirements of private enterprise data. Our study focuses on the practical deployment of Arabic LLMs in targeted applications, specifically utilizing the ACQAD (Arabic Complex Question Answering Dataset), which exhibits multi-hop reasoning. Different strategies are experimented using Long Context Window (LCW) and Retrieval Augmented Generation (RAG). Results showed that decomposing complex questions using Chain-of-Thought reasoning considerably improved the performance from 75% to 92% using LCW, but at much higher token cost compared to RAG. Trade-off between cost and performance showed that 80% accuracy can be attained using only 30% of the cost using RAG Sentence - level embeddings. Microsoft E5 embedding model is used and OpenAI GPT4-turbo LLM which proved superior reasoning performance compared to other Arabic LLMs

Keywords: Large Language Models;Retrieval Augmented Generation; Arabic NLP

1. Introduction

The advent of Large Language Models (LLMs) has markedly shifted the landscape of natural language processing, enabling a new spectrum of applications that were not possible with traditional NLP Systems before the rise of generative AI and transformers.The Reasoning capabilities of LLMs were overwhelming, and achieved significant advances in a wide range of general knowledge reasoning tasks [1]. Their reasoning power comes from pretraining of a vast amount of data [2] [3]. This data is injected as knowledge in the form of weights in the Neural-Transformer stacks. However, there are three main challenges and limitations to this knowledge representation. First, it is static, due to cut-off date and is not dynamically updated, since pretraining is a compute and cost- intensive process. Second, it fails to capture nuances and facts in specific domains. Third is the hallucination phenomenon [4].

Following the public release and subsequent acclaim of ChatGPT, there has been a marked increase in interest among corporations and enterprises. This enthusiasm has been primarily driven by the potential to deploy such tech-

E-mail address: hazem.abdelazim@eslsca.edu.eg

10.1016/j.procs.2024.10.179

nology for analyzing and leveraging their proprietary data sets [5]. The widespread admiration for ChatGPT has catalyzed a significant shift in the business community, prompting numerous organizations to explore how they might integrate this advanced technology into their operational frameworks.

Arabic, which is the main focus of the current research has received less scholarly attention compared to other languages in the current generative AI era. In terms of foundation models and instruct fined tuned LLMs, there are some reasonable effort in this area. For open source Arabic – Capable LLMs to name a few : Jais, ACEGpt, Cohere Aya, AraGPT2, ArabianGPT, and BLOOM, For proprietary Arabic LLMs there are OpenAI GPT3.5 and GPT4 with their variants as well as Claude Anthropic LLMs. Most the Models are fully Multilingual where English as the high resource language, and Arabic is a low-resource language, some models are Bilingually pretrained on Arabic and English [6], and very few are trained only on Arabic datasets like (ArabianGPT).
Despite the practical significance of evaluating Arabic large language models (LLMs) on domain-specific data—such as legal, cultural, and economic contexts—research focused on Arabic domain-specific Retrieval-Augmented Generation (RAG) implementations remains scant. While the bulk of research is directed towards foundation models and task-specific fine-tuning, implementations that utilize RAG or knowledge graphs for Arabic are exceedingly rare. This paucity of studies highlights a significant language disparity and a pronounced gap in domain-specific LLM applications in Arabic compared to other languages.

The primary focus of the current research is to explore different aspects of Arabic closed domain Question and answering simulating the enterprise requirements of reasoning across closed private data. This study addresses the practical utilization of Arabic LLMs in domain-specific applications, foregrounding the ACQAD (Arabic Complex Question Answering Dataset) [7] dataset, a recently released complex multi-hop Arabic QA dataset that requires reasoning across multiple pieces of text, and hence multi-hops to derive an answer. To best of our knowledge, this is the first research attempt exploring **Arabic** reasoning over **Multi-hop** datasets, where relatively lot of similar research has been conducted on other languages [8].

The experiments on ACQAD, involves answering a master question that requires 2-hops to get the proper answer within a context of 12 paragraphs, two of them are Gold (Ground Truth) containing the answers.The other 10 are carefully selected distractive paragraphs, which are intentionally designed to make the task challenging.Different strategies, and pipeline scenarios were investigated along two design directions. The first is whether to use LCW : Long Context Window versus RAG : Retrieval Augmented Generation by performing the embeddings on paragraph and sentence levels.The other dimension as to whether we use CoT, chain of thought reasoning to decompose the question into two subquestions, denoted by DQ or try to get the answer from the full question holistically with no question decomposition's denoted by FQ [9].

The results reached indicates interesting insights: The most powerful Arabic LLM which is OpenAI GPT4-turbo, struggled and achieved modest results of 75% accuracy, using direct full question and full context (LCW). This is due to the complex nature of th ACQAD dataset. The highest accuracy is obtained using Chain-of-Thought reasoning on LCW reaching a 92% but at a high token cost. Another insight is that there is a noticeable tradeoff between accuracy and cost (Number of tokens consumed), where 80% accuracy can be attained using only 30% of the cost using Sentence - level Embeddings. Microsoft E5 embeddings and OpenAI GPT4-turbo are used in all pipeline scenarios. The accuracy metric defined here, is the geometric mean of F1-Score and the character n-gram F-score (chrf).The next sections will cover related work with a special emphasis on Arabic language, the methodology adopted describing 6 different pipeline scenarios, followed by experimental results, economic considerations and conclusions.

2. Related work

2.1. Domain-Specific Deployments

Domain-specific deployments of LLMs is the hot subject since the resurgence of LLM in the last couple of years.Adapting LLMs to enterprise private documents, or recent and dynamic information is receiving considerable attention recently [10].Extensive research has been conducted recently on how to "inject" domain specific knowledge within the LLM stack.[4]. Four main approaches are most commonly adopted to inject new and external knowledge to LLMs:
1. **Pre-training (from scratch)** on the domain specific corpus. this approach, though looks straightforward theoretically, but is very compute, data and cost -intensive, in addition it didn't meet the expectations as the reasoning and

logical capabilities of LLMs are derived from being pretrained on a universal vast terabytes of data, beyond the size of any Domain specific data except for [11].

2. **Fine Tuning (FT)** of LLMs is another popular approach for adapting foundation models to downstream tasks. It's usually a parametric type of knowledge injection, since the main objective is to modify the model weights of the base foundation model to cater for new data or perform a particular NLP down stream task like sentiment analysis, question and answering, and summarization. The most commonly adopted FT task is instruction Supervised Fine tuning SFT, commonly referred to as chatLLMs, whether the dataset used is a structured parallel input-output, prompt-completion type of datasets.SFT is an extension to k-shot learning for a specific task. Fewer research, however, was conducted on Unsupervised Fine Tuning USFT, where the model weights are adjusted based on autoregressive next token prediction using unstructured corpus data [12].

3. **Reinforcement Learning through Human Feedback (RLHF)** to enhance the capabilities of a responsible and ethical conversational AI chat dialogue. This technique usually used as a followup training to SFT. However, both methods focus on the overall quality and conversational behavior of the AI Chat conversation rather than injecting new unseen knowledge.

4. **in-context Retrieval Augmented Generation (RAG)**. RAG is the most popular, and widely deployed approach.Simply put : using a universal pretrained LLM, and a domain specific corpus as an external knowledge source, given a user query, the relevant information is extracted and retrieved from the domain specific corpus comprising an in-context using semantic search after embedding the query as well as the text chunks which could be either a sentence or a paragraph in this research. This context is injected in a well designed prompt and applied to an LLM to generate the answer to the query The method operates on a "retrieve-and-read" framework, wherein the model accesses and utilizes content from external sources to generate the answers, thus effectively augmenting its native capabilities[13] [14] [15].

2.2. *Arabic LLMs*

The following table 1 show a summary of the most common Arabic LLMs (B*: Billion parameters), that can accept Arabic input and reply in Arabic. This is a non-exhaustive list, but the most popular as of the date of publishing this research work.

Table 1. Arabic LLMs

Model	**Architecture**	**Parameters(B*)**	**Date**	**Platform/Source**	**Pretraining Language**	**Tokenizer**
MARBERT[16]	Encoder	0.16	27-Dec-20	Open	Arabic Only	Word Piece
ARABERT[17]	Encoder	0.1	30-Mar-20	Open	Arabic Only	Word Piece
Jais[6]	Decoder	13-70	30-Aug-23	Open	Bilingual	BPE (Jais Tokenizer)
ACEgpt[18]	Decoder	7.-13	2-Apr-24	Open	Arabic Only	BPE
AraT5[19]	Encoder-Decoder	0.22	15-Mar-22	Open	Arabic Only	SentencePiece
AraGPT2[20]	Decoder	0.135 - 1.46	7-Mar-21	Open	Arabic Only	BPE
ArabianGPT[21]	Decoder	0.1 - 0.3	23-Feb-24	Open	Arabic Only	Aranizer
QARIB [22]	Encoder	0.11	21-Feb-21	Open	Arabic Only	BPE
SABER[23]	Encoder	0.369	12-Oct-22	Open	Arabic Only	BPE
ARAElectra[24]	Encoder	0.136	1-Apr-21	Open	Arabic Only	Word Piece
ARBERT	Encoder	0.16	7-Jun-21	Open	Multilingual	Word Piece
Aya-Cohere[25]	Decoder	13	12-Feb-24	Open	Multilingual	SentencePiece
GPT3.5[26]	Decoder	175	30-Nov-22	Proprietary	Multilingual	BPE
GPT4[26]	Decoder	175	14-Mar-23	Proprietary	Multilingual	BPE
Anthropic Claude[27]	Decoder	undiscolsed	15-Mar-24	Proprietary	Multilingual	BPE
Gemini[28]	Decoder	1500	May-23	Proprietary	Multilingual	BPE
Mistral (S/M/L)[29]	Decoder	undiscolsed	26-Feb-24	Proprietary	Multilingual	BPE
Qwen[30]	Decoder	1.8-7-14-72	28-Sep-23	Open	Multilingual	BPE
Mixtral (8x7B - 8-22B)[31]	Decoder	46.7-176	23-Dec-23	Open	Multilingual	BPE
Falcon 180B[32]	Decoder	180	10-Nov-23	Open	Multilingual	BPE

In our experiments we selected a subset of the aforementioned LLMs for benchmark analysis as will be described in section 4.1. The Arabic LLMs presented in the table are capable of In-context reasoning in Arabic which is a key requirement in domain-specific deployments, however, not all of them are good at universal public conversational chat in Arabic.

2.3. Arabic Embeddings

Semantic embedding plays a crucial role in RAG pipelines. By definition embeddings capture the contextual semantics of a sentence or paragraph that could be stored in vector DB for cosine similarity matching, followed be search and retrieval. For Arabic,lot of research has been conducted on word-level embeddings and relatively much fewer work on arabic-specific sentence level embedding [17]. In the current research we are referring to sentence level, not word level embeddings. Most of the sentence embedding models are for English language, and some of them are multilingual, where arabic is supported as one of the minority languages. Very few research has been conducted on benchmarks for Arabic sentence-level embeddings to select or recommend an embedding model for Arabic Language. A recent research [33], has indicated that Microsoft Multilingual E5 showed best performance among 10 state of the art multilingual embedding models, when tested on arabic dataset. Microsoft E5 embedding model is the sentence-level embedding model used in the current research. Recently OpenAI released recently ada-3-small and ada-3-large claiming superior performance on multilingual text, as well as cohere multilingual embedding.

2.4. Arabic-RAG Implementations

The RAG approach has witnessed great success in many domain specific deployments (Legal, medical, financial and many other domains). Though widely deployed for many languages, and despite considerable research has been done on Arabic Question and Answering Systems [34], Arabic RAG pipeline implementations and practical deployments are still in it's infancy and early stages, relatively. Their is a noticeable scarcity and gap, in Arabic RAG-LLM pipeline implementation [35], despite the practical impact and demand from various enterprises. The current research work is poised to open more research in this area, tackling a more complex, multi-hop reasoning approach.

3. Methodology

The study focuses on comparing different strategies and pipeline scenarios, across dimensions of performance and, cost metrics. A key dilemma in domain-specific deployments in general, as to whether use LCW : Long Context Window, i.e use the entire available text as a context to the LLM to get the answer of the query, or use RAG to extract relevant chunks for the context, to provide the required answer Specifically. The recent technological advances in LLMs is encouraging longer context windows, that may reach 128K tokens [26]and even a million token in Gemini pro [28]. This needs to be critically considered due to cost, latency and LiM phenomena (Lost in the Middle), where location of the relevant answer may be diluted in the middle for large contexts as evident in [36]. Each approach has it's pros and cons. The second stage is to implement the six possible pipeline scenarios as shown in figure 1. Each leaf in the shown figure depicts a certain pipeline scenario, for example LCW-FQ implies using full context of 12 paragraphs as a context to the LM, which the question is used holistically in full (FQ) without decomposition.DQ : implies question decomposition using chain-of-though reasoning, PE, and SE are paragraph-level and sentence-level embeddings respectively.

Central to this investigation is the identification of the most suitable RAG strategy that tackles the intricacies of Multi-Hop Arabic QA, while considering the trade-off between the performance accuracy and tokens(cost) incurred. ACQAD dataset is used throughout our experimentation which serves as a valuable resource for training and evaluating Arabic Language Understanding Models, specifically for tackling the challenges of multi-hop reasoning in complex question answering tasks as described in the next section.

3.1. ACQAD Dataset

The notion of multi-hop reasoning entails that an answer to a single question may not reside in a linear or singular segment of text but requires synthesizing information across several contexts. To date, there is no published research focusing on Arabic LLMs' efficacy in multi-hop QA. Arabic, characterized by rich morphological structure and diverse dialects, presents considerable computational costs and latency challenges when processed through LLMs, especially in comparison with English[8]. The ACQAD dataset emerges as a pivotal testing ground for Arabic RAG-LLM pipelines due to its structure, that requires complex reasoning.

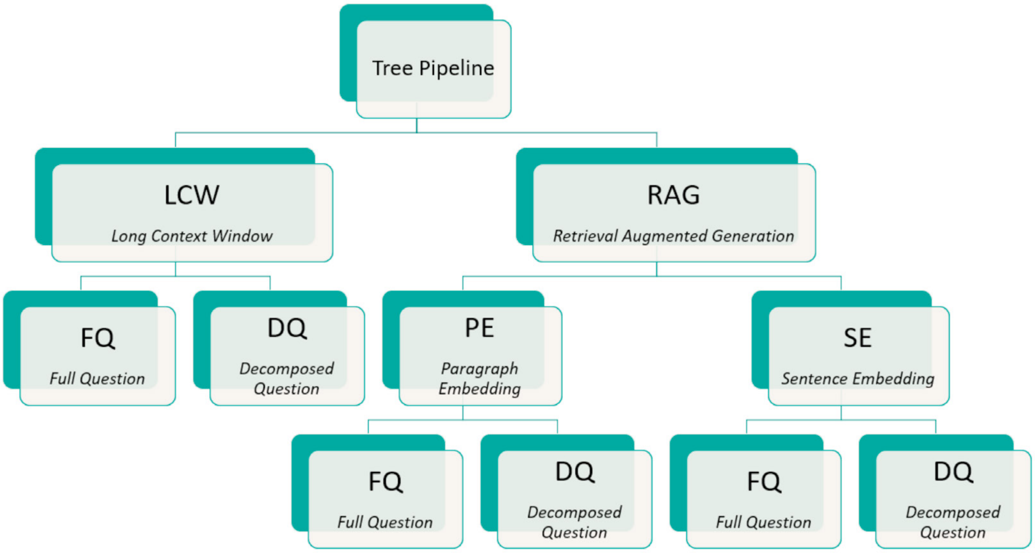
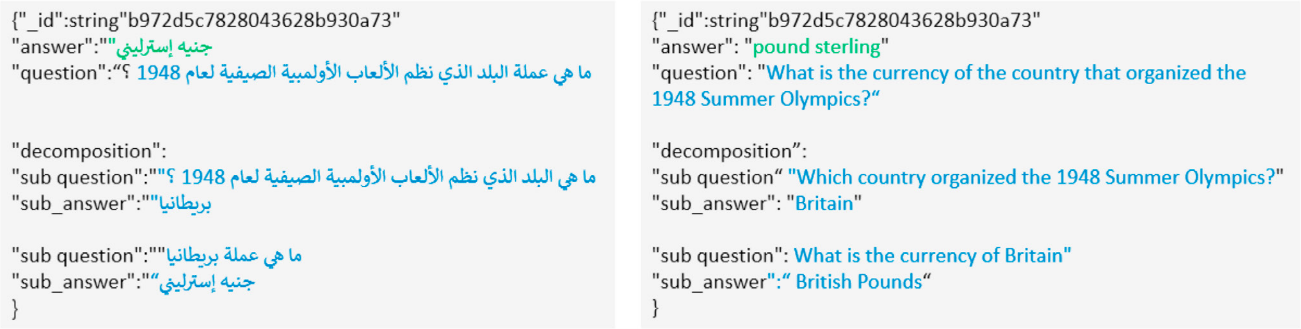


Fig. 1. Tree-pipeline : LCW-RAG different pipeline scenarios

ACQAD is a collection of multi-hop questions in the Arabic language. The dataset comprises 115k questions of comparison type and 2,984 questions of bridging type. These questions were generated from Wikipedia and ensure a wide coverage of topics. Each question-answer pair in ACQAD comes with a decomposition into single-hop sub-questions along with their corresponding answers. This feature enables researchers and developers to delve into the reasoning processes behind the complex question answering systems (explainable reasoning that imitate human reasoning). The context provided for each question consists of two gold paragraphs that contain the correct answers. Additionally, to create a realistic scenario, the dataset includes 10 distracting paragraphs that might mislead the question answering systems, so the total context for each question (LCW as defined) comprises all the 12 paragraphs.

The general structure of the dataset (JSON format) is illustrated in figure 2



```
{"_id":string"b972d5c7828043628b930a73"
"answer":""جنيه إسترليني
"question":"“ما هي عملة البلد الذي نظم الألعاب الأولمبية الصيفية لعام 1948 ؟


"decomposition":
"sub question":""ما هي البلد الذي نظم الألعاب الأولمبية الصيفية لعام 1948 ؟
"sub_answer":""بريطانيا

"sub question":""ما هي عملة بريطانيا
"sub_answer":"“جنيه إسترليني
}
```

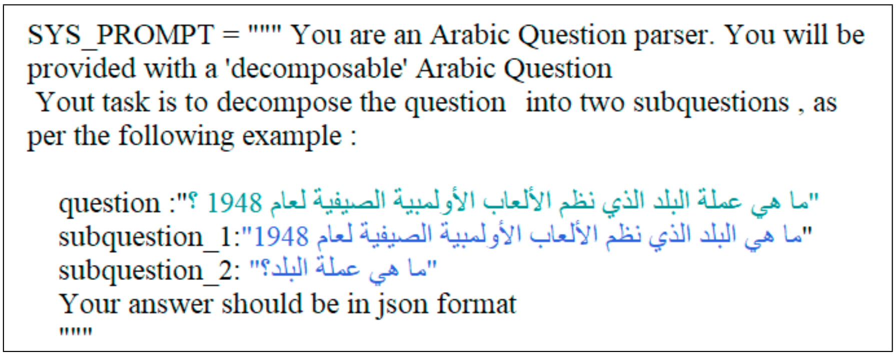
```
{"_id":string"b972d5c7828043628b930a73"
"answer": "pound sterling"
"question": "What is the currency of the country that organized the
1948 Summer Olympics?“

"decomposition”:
"sub question“ "Which country organized the 1948 Summer Olympics?"
"sub_answer": "Britain"

"sub question": What is the currency of Britain"
"sub_answer":“ British Pounds“
}
```

Fig. 2. ACQAD multi-hop Arabic example with English Translation

In the example above the main question is a complex question, requires reasoning on two different paragraphs, the answer of the first sub-question, is fed to the second sub-question to ”hop” on a different paragraph to get the final answer[7].



```
SYS_PROMPT = """ You are an Arabic Question parser. You will be
provided with a 'decomposable' Arabic Question
 Yout task is to decompose the question  into two subquestions , as
per the following example :

   question :"ما هي عملة البلد الذي نظم الألعاب الأولمبية الصيفية لعام 1948 ؟"
   subquestion_1:"ما هي البلد الذي نظم الألعاب الأولمبية الصيفية لعام 1948"
   subquestion_2: "ما هي عملة البلد؟"
   Your answer should be in json format
   """
```

Fig. 3. Chain-of-Thought question decomposition

3.2. LCW: Long Context Window

One approach to the problem is to consider the whole context, as an input to the LLM. The Long context in this case will be all the 12 paragraphs, were only two Gold paragraphs contain the correct answer. The other 10 paragraphs are carefully selected distracting paragraphs which adds more complexity to the reasoning of the LLM employed. The methodology adopted is utilizing a proper prompt to instruct the LLM to find the answer. A Typical prompt would be like

> **You are an Arabic speaker.** Given the following context and question in Arabic, find a precise Arabic answer in JSON format: {'answer':...}. Your answer should be only in Arabic.

The model answer is then compared to the ground truth (gold) answer and metrics are computed. LCW seems a natural and straightforward approach relying on the reasoning power of the LLMs however it has it's limitations as will be discussed in the experimental results.

3.3. RAG : Retrieval Augmented Generation

RAG pipeline is based on embedding certain chunks of the text provided, followed by semantic vector embedding of each text chunk. Those vectors are stored in a vector Database, in our analysis we use FAISS (Facebook AI Similarity Search). In inference time a new query is embedded using the same embedding model, and then top k similar chunks are retrieved from the vector DB using the well known cosine similarity measure. In our experiments we used Microsoft E5 embedding model and embedding on the paragraph level, i.e each paragraph is embedded in a semantic vector.

3.4. Chain of Thought

In this approach we deploy chain of thought (CoT)[9] reasoning to answer the question, a specially designed prompt using 1-shot learning as shown in figure 3. This SYS_PROMPT prompt in figure 3 decomposes the full question into two subquestions, the answer of the first subquestion is used to interrogate the LLM with a reformatted second question, to get the final answer.

3.5. Metrics

The metrics used to evaluate the performance of different pipeline scenarios are F1-score (the most commonly used metric for question answering [37],which is a function of the relying on number of tokens that are common between the gold answer and the prediction, relative to the number of tokens that are in the prediction but not in the gold answer and vice versa.

This score is widely used and well suited for European languages but suffers from clear inaccuracy for Arabic Language. we additionally employed another metric, Character n-gram F-score (chrf)[37] that is more suited for morphological rich target languages like Arabic in our case. This is depicted in the example العربية vs العربيه where the F1 score produces a zero value, as they are considered two different tokens whereas chrf produces a more realistic score of 0.84. This is very common in Arabic Language where both tokens are considered semantically correct.The overall combined accuracy used is the **geometric mean** of F1-score and chrf, which accounts for the both exact words matching and character n-grams matching to factor-in the morphological and inflectional nature of Arabic generated responses.

4. Experimental Results

4.1. Reading Comprehension

We embark on a comparative benchmark analysis of 100 random samples from the ACQAD dataset, carefully examining the capabilities of various LLMs in a reading comprehension setting. In this experiment we bench-marked 7 popular Arabic LLMs, 3 proprietary (GPP3.5,4, and Gemini) and 7 open source models. The purpose of this experiment is to set the baseline for the next set of experiments comprising more complex reasoning using different pipeline scenarios. This reading comprehension task is a much simpler task compared to Multi-hop QA where all LLMs are provided with the Gold paragraph directly against a set of sub-questions, and compare the model answer with the ground Truth.

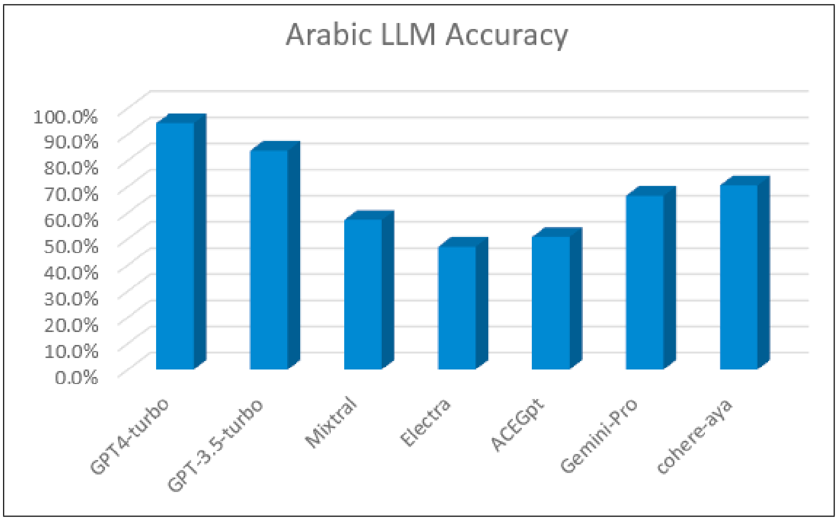


Fig. 4. Arabic LLM Reading Comprehension Accuracy

The results shown in Figure 4, indicates the superior performance of openAI proprietary models, namely openAI GPT 4.0 reaching 94% followed by GPT3.5,which indicates the reasoning power of proprietary models over open source models on Arabic language reasoning

4.2. Different pipeline scenarios

In this experimental setup we benchmark all the six pipeline scenarios explained in section 3. We used the top performant model in the base line reading comprehension experiment which is GPT4-turbo, and Microsoft E5 embedding for paragraph and sentence embedding. Two major directions were experimented, namely LCW: Long context Window, and RAG: Retrieval Augmented Generation, as follows:

4.2.1. LCW : Long Context Window

In this setting we used the Long context Windows (LCW), i.e all the 12 paragraphs containing around a million token (939000 tokens) are used as one context to the LLM prompt. Two scenarios are tested, first using the full

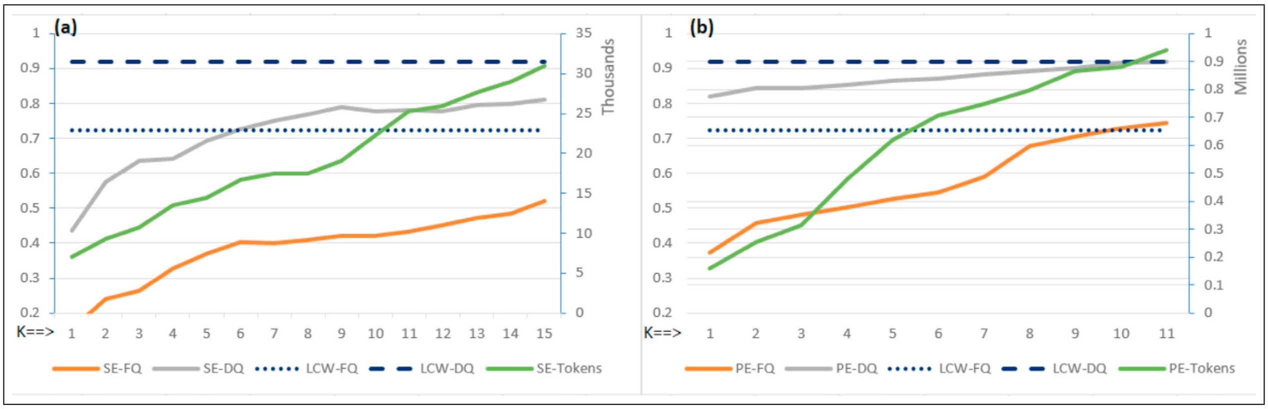


Fig. 5. (a) RAG Sentence Embedding (b) RAG Paragraph Embedding

complex question (FQ) and second using Decomposed Question (DQ), by applying chain of thought reasoning. In the the full question (FQ) scenario as shown in **figure 5**. The LCW-FQ is displayed as the horizontal dotted line, and LCW-DQ is the horizontal dashed line. The primary vertical axis on the left is the accuracy as defined in section, the secondary vertical axis on the right shows the tokens consumption.Despite the superior reasoning power of GPT4, which is considered the world's most capable LLM[38], it gave an accuracy of 74% as seen in the figure using full question denoted by LCW-FQ. The main reason for that relatively low acc challenged by ”distractive” paragraphs in the ACQAD Dataset, so GPT4 in many cases could not reason properly on multi-hop paragraphs inside the Long Context Window. When we applied chain of thought reasoning to decompose the question to two relevant subquestions (DQ) the results were much superior reaching 92% (as denoted by LCW-DQ). which is a great result, however the LCW approach in general is costly as it consumes large number of tokens ($) per question which may be unfeasible practically in particular for the Arabic language in practical enterprise deployment implementations.

4.2.2. RAG: Retrieval Augmented Generation

In this scenario we used embedding either on the paragraph level (PE) or the sentence level (SE)using Microsoft E5 Multilingual embedding and FAISS vector DB. The results shown in figure 5 are the accuracies for paragraph embedding (PE) for incremental values of K which is the number of top hits resulting from the vectorDB semantic search. in figure 5(a) K refers to the top ”sentence” hits, while in figure 5(b), K refers to the top ”paragraph” hits.

Obviously the accuracy is a monotonic increasing function of K, as more context means more chance to get the correct answer, and it's evident that question decomposition,denoted by PE-DQ gave better accuracy in the range of 30%-40% over the full question denoted by PE-FQ, indicating the importance of chain-of-thought reasoning when dealing with complex questions of multi-hop nature. As K approaches 12 this will be identical to the LCW scenario.

Figure 5(a) shows the same analysis using sentence embeddings (SE) which is expected to be very economic in terms of token consumption Also results clearly depict the superiority of DQ (Decompose Question) over using full question (FQ). An important insight from, is that we can reach 80% accuracy @K=15 using 30000 tokens which means saving around 70% of the cost, as compared to paragraph embedding (PE).Also, diminishing returns are observed as K increases above 9 sentence hits.

5. Discussion : Economic considerations and Tokenization

The previous analyses showed that for large scale practical deployment of domain specific application, using proprietary powerful models like OpenAI GPT4.0 and GPT3.5 is the current pragmatic choice due to their high performance, as of to-date.However proprietary models charge the user based on tokens consumed which is a function of the context size. This implies that economic considerations in terms of $ costs is of essence. One main problem with minority languages in LLM pre-training is the tokenization algorithm, since the cost of APIs consumption is driven by the number of input and output tokens. for example as per the date of this research, the cost of input tokens in

GPT4-turbo is 10$ per million token, and 30$ per million output tokens, so the more tokens consumed the more cost incurred. In Arabic Language the tokens are more or less one character, as for English the token is a word or part of a word.So on average the cost for Arabic enterprise deployments will be 3-3.5 times the corresponding English cost [39]. This is due to the nature of the BPE Byte-pair encoding statistical tokenizers which favours high resource languages over minority low resource languages. Having said that, the cost consideration is more pressing for Arabic large scale deployments which strengthens our arguments in this research to consider cost-accuracy trade-off when designing such systems.

6. Summary, Conclusion and Future research

This research examined the application and effectiveness of Large Language Models (LLMs) for Arabic closed-domain question and answering systems, focusing on the ACQAD dataset. The study revealed profound capabilities of generative AI in processing and understanding complex Arabic queries, leveraging advanced techniques such as Long Context Window (LCW) and Retrieval Augmented Generation (RAG). Experiments demonstrated the utility of Chain-of-Thought (CoT) reasoning, which significantly enhanced performance by decomposing complex questions into manageable sub-questions. Results showed that decomposing complex questions using Chain-of-Thought reasoning considerably improved the performance from 75% to 92% using LCW, but at much higher token cost compared to RAG. A strategic trade-off was identified, where reasonable accuracy (80%) could still be achieved at only 30% of the cost by utilizing RAG Sentence-level embeddings pipeline.

The study highlighted key insights into deploying Arabic Large Language Models (LLMs) for domain-specific applications, emphasizing that the Chain-of-Thought reasoning approach is the most effective for complex Arabic QA tasks. It revealed a trade-off between computational cost and accuracy, stressing the importance of cost-efficiency, especially in enterprise environments. The research provided benchmarks for Arabic LLMs in reading comprehension and reasoning, offering guidance for future model selection. It recommended a tiered approach to question answering to balance performance and cost and called for ongoing research and development to address the unique challenges of the Arabic language, particularly in domain-specific implementations and the exploration of new embedding models and fine-tuning approaches.

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, and S. Agarwal, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[2] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel, "Language models as knowledge bases?" in *Conference on Empirical Methods in Natural Language Processing*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:202539551

[3] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, and J. Li, "A survey of knowledge enhanced pre-trained language models," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[4] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *arXiv preprint arXiv:2311.05232*, 2023.

[5] W. Yu, W. Yu, C. Zhu, Z. Li, Z. Hu, Q. Wang, H. Ji, and M. Jiang, "A survey of knowledge-enhanced text generation," *ACM Computing Surveys*, vol. 54, pp. 1 – 38, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:222272210

[6] N. Sengupta, S. Sahu, B. Jia, S. Katipomu, H. Li, F. Koto, O. Afzal, S. Kamboj, O. Pandit, R. Pal, and L. Pradhan, "Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models," *arXiv preprint arXiv:2308.16149*, 2023.

[7] A. Sidhoum, M. Mataoui, F. Sebbak, and K. Smaïli, "Acqad: a dataset for arabic complex question answering," in *International conference on cyber security, artificial intelligence and theoretical computer science*, 2022, December.

[8] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[9] B. Wang, S. Min, X. Deng, J. Shen, Y. Wu, L. Zettlemoyer, and H. Sun, "Towards understanding chain-of-thought prompting: An empirical study of what matters." Association for Computational Linguistics, 2023.

[10] W. Lan, Y. Chen, W. Xu, and A. Ritter, "An empirical study of pre-trained transformers for arabic information extraction," in *Conference on Empirical Methods in Natural Language Processing*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:226262317

[11] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "Bloomberggpt: A large language model for finance," *ArXiv*, vol. abs/2303.17564, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257833842

[12] O. Ovadia, M. Brief, M. Mishaeli, and O. Elisha, "Fine-tuning or retrieval? comparing knowledge injection in llms," *ArXiv*, vol. abs/2312.05934, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266162497

[13] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. Van Den Driessche, J. Lespiau, B. Damoc, A. Clark, and D. de Las Casas, "Improving language models by retrieving from trillions of tokens," in *International conference on machine learning*. PMLR, 2022, June, pp. 2206–2240.

[14] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Generalization through memorization: Nearest neighbor language models," *arXiv preprint arXiv:1911.00172*, 2019.

[15] B. Wang, W. Ping, L. McAfee, P. Xu, B. Li, M. Shoeybi, and B. Catanzaro, "Instructretro: Instruction tuning post retrieval-augmented pretraining," *arXiv preprint arXiv:2310.07713*, 2023.

[16] M. Abdul-Mageed, A. Elmadany, and E. Nagoudi, "Arbert & marbert: Deep bidirectional transformers for arabic," *Annual Meeting of the Association for Computational Linguistics*, 2020.

[17] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, H. Al-Khalifa, W. Magdy, K. Darwish, T. Elsayed, and H. Mubarak, Eds. Marseille, France: European Language Resource Association, May 2020, pp. 9–15. [Online]. Available: https://aclanthology.org/2020.osact-1.2

[18] H. Huang, F. Yu, J. Zhu, X. Sun, H. Cheng, D. Song, Z. Chen, A. Alharthi, B. An, Z. Liu, and Z. Zhang, "Acegpt, localizing large language models in arabic," *ICLR 2023 Conference,*, 2023.

[19] A. Elmadany and M. Abdul-Mageed, "Arat5: Text-to-text transformers for arabic language generation," in *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long papers)*, 2022, May, pp. 628–647.

[20] W. Antoun, F. Baly, and H. Hajj, "Aragpt2: Pre-trained transformer for arabic language generation," *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, 2020.

[21] A. Koubaa, A. Ammar, L. Ghouti, O. Necar, and S. Sibaee, "Arabiangpt: Native arabic gpt-based large language model," *arXiv preprint arXiv:2402.15313*, 2024.

[22] A. Abdelali, S. Hassan, H. Mubarak, K. Darwish, and Y. Samih, "Pre-training bert on arabic tweets: Practical considerations," *arXiv preprint arXiv:2102.10684*, 2021.

[23] A. Ghaddar, Y. Wu, S. Bagga, A. Rashid, K. Bibi, M. Rezagholizadeh, C. Xing, Y. Wang, X. Duan, Z. Wang, B. Huai, X. Jiang, Q. Liu, and P. Langlais, "Revisiting pre-trained language models and their evaluation for Arabic natural language processing," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 3135–3151.

[24] W. Antoun, F. Baly, and H. Hajj, "Araelectra: Pre-training text discriminators for arabic language understanding," *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, 2020.

[25] A. Üstün, V. Aryabumi, Z. Yong, W. Ko, D. D'souza, G. Onilude, N. Bhandari, S. Singh, H. Ooi, A. Kayid, and F. Vargus, "Aya model: An instruction finetuned open-access multilingual language model," *arXiv preprint arXiv:2402.07827*, 2024.

[26] OpenAI, J. Achiam, and S. A. et al., "Gpt-4 technical report," 2024.

[27] Anthropic, "The claude 3 model family: Opus, sonnet, haiku," 2024, preprint.

[28] e. a. Gemini Team Rohan Anil, "Gemini: A family of highly capable multimodal models," 2024.

[29] A. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. Chaplot, D. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, and L. Lavaud, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[30] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, and B. Hui, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[31] A. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. Chaplot, D. Casas, E. Hanna, F. Bressand, and G. Lengyel, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024.

[32] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, Étienne Goffinet, D. Hesslow, J. Launay, Q. Malartic, D. Mazzotta, B. Noune, B. Pannier, and G. Penedo, "The falcon series of open language models," 2023. [Online]. Available: https://arxiv.org/abs/2311.16867

[33] H. Abdelazim, M. Tharwat, and A. Mohamed, "Semantic embeddings for arabic retrieval augmented generation (arag)," *International Journal of Advanced Computer Science & Applications*, 2023.

[34] A. Alrayzah, F. Alsolami, and M. Saleh, "Challenges and opportunities for arabic question-answering systems: current techniques and future directions," *PeerJ Computer Science*, vol. 9, p. e1633, 2023.

[35] A. Mahboub, M. Za'ter, B. Alfrou, Y. Estaitia, A. Jaljuli, and A. Hakouz, "Evaluation of semantic search and its role in retrieved-augmented-generation (rag) for arabic language," *arXiv preprint arXiv:2403.18350*, 2024.

[36] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the Middle: How Language Models Use Long Contexts," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 02 2024.

[37] M. Popović, "chrF: character n-gram F-score for automatic MT evaluation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, C. Hokamp, M. Huck, V. Logacheva, and P. Pecina, Eds. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 392–395. [Online]. Available: https://aclanthology.org/W15-3049

[38] Artificial Analysis AI, "Artificial analysis ai," 2023, accessed: 2023-07-01. [Online]. Available: https://artificialanalysis.ai/

[39] O. Ahia, S. Kumar, H. Gonen, J. Kasai, D. R. Mortensen, N. A. Smith, and Y. Tsvetkov, "Do all languages cost the same? tokenization in the era of commercial language models," 2023. [Online]. Available: https://arxiv.org/abs/2305.13707