Enhancing Arabic Retrieval Augmented Generation through Language Processing

Shadi Saleh[1], **Belkacem EL Jattari**[2], **Layth Oud**[1], **Maryam Alblooshi**[1], **Bouthaina Lakhdari**[2], **Ali Alnaqbi**[1],
[1]Prime Technologies, Czechia,
[2]Mohammed bin Zayed University for Humanities, Humanities and Social Sciences
Correspondence: shadi@prime-itech.com

Abstract

We investigate the impact of text preprocessing on Arabic Information Retrieval (IR) systems and, consequently, on the quality of Retrieval-Augmented Generative (RAG) systems. Our work focuses on academic content in Arabic. We analyze how the IR performance affects the quality of RAG systems in answering users' questions on various academic topics. Our findings indicate that the performance of an IR system is significantly influenced by the quality of Optical Character Recognition (OCR) applied to PDF files. We employ a state-of-the-art deep learning-based OCR solution to create our IR index. Eventually, this IR index is used to generate a context-window for the generative model that is employed in chat assistant to answer questions in the scientific domain. We introduce a benchmark dataset for the IR system, comprising 170 Arabic queries and IR relevance assessment with numerous query-document judgment pairs. Our results demonstrate that advanced text preprocessing can lead to an increase of 8 points in terms of $P@5$ of the IR mode, an increase of 11% in the accuracy of the answering system, and up to 95% of correct citations compared to our baseline system.

1 Introduction

The Arabic language presents unique challenges for Natural Language Processing (NLP) applications, particularly in Information Retrieval (IR), Question Answering (QA), and Optical Character Recognition (OCR). These challenges stem from the script's complexity, including its cursive nature, contextual letter forms, and bidirectional text orientation. Additionally, the scarcity of high-quality annotated datasets for Arabic further exacerbates these issues. In recent years, Retrieval-Augmented Generation (RAG) systems have emerged as a promising approach to enhance generative models like GPT by grounding their outputs in relevant retrieved documents. This paradigm has proven effective in reducing hallucinations and improving factual accuracy by providing contextually relevant information to generative models.

This paper investigates the impact of advanced Arabic preprocessing techniques on IR systems and their downstream influence on RAG performance. Specifically, we focus on academic content from Arabic scientific papers published in PDF format where accurate OCR and IR are critical for generating high-quality responses. By employing a state-of-the-art deep learning-based OCR solution and optimizing the IR pipeline, we demonstrate significant improvements in both retrieval precision and the relevance of generated answers. Our contributions include a comprehensive evaluation of OCR tools for Arabic PDF processing, and their impact on IR performance, and the development of an Arabic RAG system capable of answering academic queries with enhanced relevance and citation accuracy.

2 Related Work

Recently, RAG systems have attracted significant attention from researchers tackling various downstream tasks, particularly in the development of customized chatbots. RAG combines the strengths of retrieval and generative models, enabling chatbots to access grounding knowledge-base and produce contextually relevant responses. This approach has proven effective in creating chatbots that are tailored to in-domain data, by building a retrieval system that searches within in-domain data index and provides context to generative model which has shown to significantly reduces hallucination and provides timely updated information. The ability of generative models to synthesize information from diverse sources showed recently that such models can be used to achieve comparable results to state-of-the-art methods in various NLP tasks (Brown et al., 2020), such as translation, translation-quality evaluation (Kocmi and Federmann, 2023), and question-answering, usually applying generative models in down-stream tasks requires a few-shot learning approach that can be applied post model training via system prompt.

Sadek et al. (2012) presented one of the first Arabic question-answering system designed to answer *why* and *how* questions. Their approach relies on the discourse structure of Arabic texts to automatically find answers. This method uses Rhetorical Structure Theory (RST), which has been proven effective in NLP applications.

Studies indicated that off-the-shelf OCR tools, which perform considerably well on English, struggle with Arabic due to the script's complexity and the scarcity of high-quality labeled datasets for training purposes. Such models often utilize Hidden Markov Model for sequence modeling (Bunke et al., 1995) . In order to achieve reasonable performance, more advanced method including leveraging language modeling and deep learning approaches (Bhatia et al., 2024).

IR-RAG @ SIGIR24 is a dedicated workshop in SIGIR that emphasizes on the critical rule of IR systems as an internal component of RAG system (Petroni et al., 2024). Multiple submissions argued that the effectiveness of RAG systems heavily relies on the quality of retrieved documents, as poor-quality or irrelevant sources can lead to misleading outputs, and called for a further exploration of robust retrieval mechanisms to enhance RAG capabilities [1].

3 Experiment Settings

3.1 Data

3.1.1 Queries

We leverage multiple data sources to enhance our analysis. Our dataset comprises 12,000 PDF files of Arabic journal papers sourced from the Shamra Academia portal[2]. This portal is accessible online and indexed by major search engines such as Google[3] and Bing[4]. These search engines provide a search console for website owners, offering insights into visitor engagement, clicks, and search queries.

We analyzed the query logs from the past six months, focusing on the top 100 most frequently searched queries each month. This approach yielded a total of 600 queries covering various academic research topics. To ensure a diverse and comprehensive set of queries, three experts (native Arabic speakers and academic researchers) selected 170 queries from this pool. We split these queries into 100 queries for training and 70 queries for testing. The split takes into account preserving the distribution of query categories. Table 1 shows the distribution of query categories on both the training and test set. Queries were manually segmented into 8 categories, based on the field of the study related to researches that are relevant to each query.

Category	train queries	test queries
Economy	14	10
History	11	8
Engineering	15	11
Agriculture	14	10
Science	6	4
Math	4	3
Medicine	10	7
Literature	26	18

Table 1: Distribution of query categories on both training and test sets

Table 2 shows a sample of query text and their categories. The query text is what users type in the search engine to find information that meets their information need. It is important to mention that query text is not necessarily a good representation of searcher's information need. When searchers need to find information about a specific topic, they predict how an ideal document will look like, and what might be the main keywords in that document (e.g. the document title, headers and sub-headers), and they expect the IR system to find those documents for them (El Zein and da Costa Pereira, 2022).

Category	**Query**
Literature	إدارة التغيير التربوي
Engineering	إدارة البيانات السحابية
Medicine	التصلب اللويحي
History	منهج التأليف في السنة النبوية

Table 2: Samples of queries chosen and proofread by expert annotators

[1] https://ceur-ws.org/Vol-3784/

[2] https://shamra-academia.com

[3] https://google.com

[4] https://bing.com

3.1.2 Documents

We have access to 12,000 documents in the portal. These documents are academic papers published in various open-access journals with which the portal has agreements.

Each document in the portal has the following entries (in addition to the PDF file): title, abstract, keywords and the main content. Those fields were added by authors and verified by the portal editors. However, there is still large amount of text in the PDF files needs to be extracted and added to the index so it can be searchable (the main content). To extract the text from PDF files, we experiment with two methods: The first method is based on an open-source python library PyPDF [5], and the second method is based on a more advanced method that utilizes Deep Learning (Surya)[6].

Table 3 presents a sample sentence extracted from a PDF file after conversion to text using two different libraries. The PyPDF library consistently exhibits issues such as merging two words by eliminating the space between them or omitting letters from words. This results in significant information loss. In the sample shown, out of 15 words, PyPDF correctly extracted only 6 words. In contrast, the Surya library accurately extracted the entire sentence without any error. These findings were consistently observed across other documents as well. The primary reasons for PyPDF's poor performance with Arabic text are the complexity of the Arabic script and the bidirectional nature of the language. Arabic script includes cursive writing, contextual letter forms, and diacritics, which are challenging for OCR systems primarily designed for Latin scripts like English. Additionally, Arabic is written from right to left, adding another layer of complexity that PyPDF may not handle effectively.

We observed some issues with Surya OCR system as well when handling Arabic terms that are in-domain, e.g. scientific terms in the medical and engineering domains. To further improve the output of the Surya OCR system and restore information loss, we leverage an LLM as an auto-correction system, where we ask the LLM to improve the output of OCR using the following prompt:

> *You are an expert copyeditor specializing in academic and scientific Arabic texts. Your task is to correct errors in a given OCR-scanned paragraph, consider the following:*
>
> ***Text Source:*** *The input is from an Optical Character Recognition (OCR) system. Therefore, you must identify and correct common OCR errors, which include spelling mistakes, garbled words, and incorrect character recognition.*
>
> ***Content Type:*** *The text is academic and contains specific scientific terminology. You must preserve all original technical terms and academic phrases without alteration or simplification.*
>
> ***Correction Scope:*** *Your corrections should focus exclusively on spelling, grammar, and reconstructing garbled words to make the text fluent and accurate. Do not rewrite, rephrase, or change the intended meaning of the original content.*
>
> ***Output Format:*** *Provide ONLY the fully corrected Arabic text. Do not include any introductory phrases, explanations, or commentary.*
>
> ***Text to be corrected:***
> *{{paragraph_input}}*

From observations with preliminary prompt designs, it was noted that the LLM occasionally tended to rephrase the input text or introduce additional words, altering the original meaning. To remedy this and address common OCR-related errors such as character misrecognition and garbled words, the proposed LLM prompt explicitly identifies the text as OCR output and is designed to instruct the model to preserve domain-specific terminology while strictly avoiding any rephrasing or insertion of extra words.

3.1.3 Annotations

To evaluate the performance of our IR system, we conducted a human evaluation using the open-source tool: *relevation*[7]. Relevation is a web application in which human judges can evaluate the relevance of the retrieved documents.

We asked three human judges, all are academic researchers and native Arabic speakers, to perform the evaluation. Each judge was presented with a query and a corresponding document in PDF format. Alongside the query, we provided the information need behind it, and the judges' task was to determine whether the document satisfied the

[5] https://github.com/py-pdf/pypdf

[6] https://github.com/VikParuchuri/surya

[7] https://github.com/ielab/relevation

OCR System	Sentence
PyPDF	تقييمحدوث الإستنشاق الربي لدىمرضى العناية الحرحة الموضوع ليم التغذية المعوة بالانبوب الأفي المعدي
Surya	تقيم حدوث الإستنشاق الرئوي لدى مرضى العناية الحرجة الموضوع لهم التغذية المعوية بالانبوب الأنفى المعدى

Table 3: Samples of sentence taken from a PDF file of a paper published in a medical journal. The sentence is taken after converting the PDF file into text using two OCR librares, PyPDF

user's need. We employed a ranked evaluation approach, where each document can be annotated as highly relevant, relevant, somewhat relevant, or irrelevant based on how well it addresses the user's information need. At the end of the evaluation, we analyzed the agreement rate among the judges. Each judge was shown 30 document-query pairs that had been previously judged by another judge, and they were asked to re-evaluate these pairs. This process provided insights into the reliability of the judgments. We then binarized the judgments into a 0-1 scale, where irrelevant and somewhat relevant were coded as 0, and relevant and highly relevant as 1. Before binarization, the agreement rate was 70%, which increased to 78% after binarization. This increase is understandable given the varying degrees of relevance perceived by the judges, with the highest disagreement occurring between the somewhat relevant and relevant degrees.

3.2 Indexing

To build the IR system that is used to inject context to the LLM prompt, we employed Elasticsearch to retrieve the top 100 most pertinent studies for each query in our dataset. We constructed two distinct indexes for this purpose: one utilizing PyPDF and the other based on Surya. The search filter was designed to optimize the relevance of the results, and include all possible information in each document. The Elasticsearch query schema used is as follows:

```
{
    "multi_match": {
        "query": "query goes here",
        "fields": [
            "title^3",
            "abstract^2",
            "content"
        ],
        "type": "best_fields",
            "tie_breaker": 1
        }
    }
```

Filter Schema: The following is the filter schema that is used to query the ElasticSearch index:

- **multi_match**: This is a query type in Elasticsearch that allows searching across multiple fields. It is particularly useful when the same query needs to be matched against various fields in the documents.
- **query**: Represents the search terms or phrases written by the user. This is the text that Elasticsearch will look for across the specified fields.
- **fields**:
    - **"title^3"**: This field corresponds to the full title of the document in Arabic. The ^3 denotes a boost factor of 3, meaning matches in the title are considered three times more relevant than matches in fields without a boost.
    - **"abstract^2"**: This field contains the abstract of the document in Arabic. With a boost factor of 2, matches here are deemed twice as relevant compared to unboosted fields.
    - **"content"**: This field encompasses the main content of the research documents, extracted from PDFs using either PyPDF or Surya. It does not have an explicit boost factor, so it serves as the baseline relevance.
- **type**: Set to **"best_fields"**, this parameter instructs Elasticsearch to return documents that have the best score from any one field. It focuses on the single most relevant field match

rather than combining scores from multiple fields.

- **tie_breaker**: With a value of **1**, the tie breaker adjusts the scoring when documents match the query in multiple fields. A higher tie breaker increases the influence of secondary matches on the overall score, ensuring that documents with multiple field matches are ranked higher.

More information about Elasticsearch Query Language can be found in the official documentation [8].

By leveraging this schema, we prioritized documents where the query terms appeared in the title or abstract, reflecting a higher likelihood of relevance. The boosting factors ensured that matches in the title and abstract had a more significant impact on the relevance score than matches in the main content. The use of the **best_fields** type, combined with an appropriate tie breaker, allowed for a balanced and effective retrieval of documents, enhancing the quality of the search results; after retrieving the search results from Elasticsearch, we extracted the relevance scores and the ranks of the documents as provided by the search engine. These scores and ranks are fed into the *relevation* tool for human evaluation.

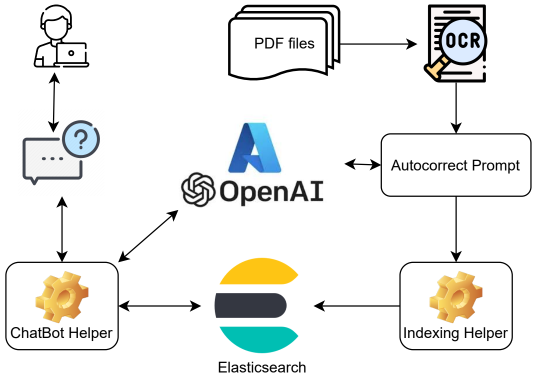

Figure 1: System architecture of the proposed chatbot, illustrating the interaction between the IR, the LLM (OpenAI's GPT-4-turbo) and the auto-correct components for generating accurate, context-aware responses.

3.3 QA Evaluation

The classical way of evaluating QA systems requires gold dataset, which includes a set of questions and their answers. Then usually a metric is based on lexical matching between the system-generated answer (prediction) and the real answer (reference) is used to generate a similarity score. Kamalloo et al. (2023) has shown that lexical-matching approaches suffer from a big fall, because of two reasons:

- It is almost impossible to provide a list of gold answers that cover all possibilities.
- Lexical matching methods cannot detect hallucinations and false claims in the output of LLM models.

Another work by Min et al. (2021) demonstrated that when humans who have experience in the domain evaluate the performance of question-answering (QA) system, the evaluation metric can increase up to 23% compared to evaluations conducted through automated methods. Encouraged by their finding, we choose to perform human-evaluation of our proposed RAG system.

4 Experiments and Discussion

Figure 1 shows our system architecture. The system consists of mainly two components: the IR component and the Chatbot component. Users start their sessions by posing a question to the system, then the ChatBot helper generates queries from this question and conducts a search from the index. The purpose of the search is to build up context for the LLM model that will be used to generate a full answer to the user's question. This context helps reduce hallucination and makes inferring answers accurate by providing scientific references for users. The context consists of a list of papers (title, abstract and content) that will be injected to the system prompt. For that reason, a relevant context is highly important to generate answers that satisfy the user's question. To ensure high quality context, focus mainly on tuning choosing IR system that performs reasonably well for our use-case.

When indexing a new document, the following process is initiated: First, the document is parsed using an OCR system to extract its text. This text is then passed to the LLM with the proposed auto-correction prompt to fix any potential error introduced by the OCR. Finally, the corrected text is indexed by Elasticsearch, making it available for search queries.

[8] https://www.elastic.co/blog/getting-started-elasticsearch-query-language

4.1 IR experiments

To experiment with multiple scoring models in our dataset, we evaluate the performance of the IR system against the training dataset on the following retrieval model: LM-Dirichlet (Blei et al., 2003), Okapi BM25 (Crestani et al., 1998), and LM-JelinekMercer (Zhai and Lafferty, 2017).

A list of retrieval models supported by ElasticSearch is available in the official documentation[9]. We focus in our IR evaluation on two IR metrics: Normalized Discounted Cumulative Gain (*NDCG@k*), and Precision at k (*P@k*). *NDCG@k* considers both the relevance and the position of the retrieved documents, with higher weights assigned to the results at the top of the list. *P@10* measures the proportion of relevant items in the top k ranked documents, focusing solely on precision without considering the ranking order within the top-k items (Teufel, 2007).

Our IR experiments aim to address two pivotal questions:

- Can an off-the-shelf OCR model deliver satisfactory performance for Arabic IR, or is there a need for a more sophisticated model?
- Which IR model demonstrates superior performance in our experimental settings?

To address these two questions, we construct two ElasticSearch indices: one utilizing text extracted from PDF files using PyPDF library, and another employing the Surya model. The details of these indices are elaborated in Section 3.2. There is no difference between these two indices nor the querying mechanism.

Table 4 shows the evaluation results for the three IR models against the test set. This set is unseen and was not used to make any decision. The results show consistent improvement of both NDCG@5 and P@5 for all retrieval models when using more advanced OCR system (Surya), the biggest improvement is shown in Okapi BM25 model, where $NDCG@5$ increased by $+4.85$ points, and $P@5$ by $+7.53$ points. It is evident that Okapi BM25 demonstrates superior overall ranking quality ($NDCG@5$), LM-JelinekMercer shows slightly better precision in the top 5 results ($P@5$). This suggests that Okapi BM25 might be more effective at distinguishing between degrees of relevance across the result set, while LM-JelinekMercer is particularly good at identifying highly relevant documents for the top positions, possibly due to its smoothing technique being well suited to the characteristics of Arabic scientific text. Considering those results, we decide to choose Okapi BM25 as the scoring model for our IR system, and Surya model as an OCR system to parse the PDF files.

4.2 Chatbot

In this section, we discuss how we use our IR system and leverage the LLM model to answer users' questions. First, user starts their session by posing a question to the system, as shown in Figure 1. Then the question is used as a query to retrieve top 5 relevant documents from Elasticsearch. Each document includes its title, abstract and content. These documents are then injected into the prompt, then a request is made to the Azure OpenAI API chat completion endpoint (*GPT-4 turbo-2024-04-09* version) [10] to answer the user's question. The LLM prompt is as follows:

> *You are an Expert Academic Research Synthesizer. Your function is to act as a research assistant, tasked with extracting and synthesizing information exclusively from a provided corpus of scientific documents. Strictly follow these instructions:*
> *1.* ***Corpus Definition****: You will receive a series of academic papers, each formatted with a clear title and content.*
> *2.* ***Information Adherence****: You MUST NOT use any external knowledge. All information in your response must be directly derived from the provided papers.*
> *3.* ***Answer Structure****: The response must be a comprehensive and cohesive synthesis of the information relevant to the question. Do not provide a list of facts; instead, integrate findings from multiple papers to create a single, detailed answer.*
> *4.* ***Specificity and Detail****: Focus on providing an extremely specific and factual answer. Avoid all forms of vague, generic, or abstract language.*

[9] https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules-similarity.html

[10] https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models

*5. **Citations**: Every single statement of fact or claim must be followed immediately by a citation in the format '(PaperTitle)'. If a sentence synthesizes information from multiple papers, list all relevant citations.*
*6. **Language**: The final, complete answer MUST be generated entirely in the Arabic language.*
7. Provide only the answer to the question. Given the following papers:
{{Title Abstract Content}}
.
.
{{Title Abstract Content}}
Answer the following question: {{Question}}

The prompt is designed to make the LLM simulate the process of a human researcher while strictly preventing hallucination. By forbidding external knowledge and demanding that every factual statement be anchored to a source with a mandatory citation, we enforce a high degree of verifiability, ensuring the model cannot invent information. This mimics a researcher's reliance on primary sources. Moreover, the instruction to integrate findings from multiple papers to create a single, detailed answer compels the model to move beyond simple fact extraction and replicate the human cognitive process of synthesis. This dual approach ensures that the generated output is not only factually grounded and trustworthy but also demonstrates a sophisticated, human-like understanding of the source material, a crucial requirement for reliable academic use.

To evaluate the performance of our proposed Chatbot architecture, we developed two Telegram bots utilizing the Telegram Bot API [11]. This API allows for the creation of programs that use Telegram messages as an interface. Users can interact with it using their mobile devices or Telegram desktop version. Our experimental setup consisted of two distinct bots:

- **Baseline Bot** This bot directly sends user questions to the LLM endpoint. The only system prompt that we use in the baseline system is: *Answer the following question and provide citations for your answer*: $\{question\}$.
- **The Proposed Chatbot (ArabicRAG)** This bot implements our proposed architecture (as described in Figure 1), utilizing the prompt we presented earlier, including contextual information.

Using the Telegram API, we were able to create a robust experimental framework to compare the performance of our proposed architecture with the baseline system. We asked our human judges to send each question in the test set to both systems and give each answer a grade based on its relevance. We introduce the following grading system, the final grade is the sum of all of them:

- **Does the generated output provide correct citations?**
  - 3: There is at least one correct citation for each statement.
  - 2: Some correct citations are missing, but not very crucial.
  - 1: Crucial citations are missing, or incorrect citations are provided.
  - 0: There are no correct citations provided.
- **Does the generated output answer your question?**
  - 3: Yes, the output fully answers my question.
  - 2: The output partially answers my question.
  - 1: The output somewhat answers my question.
  - 0: The output does not answer my question.

Then we take the average of the two grades, a perfect answer will be graded 3 for citation and 3 for correctness, yielding an average of 3 final grade. This evaluation is done for the 70 queries in the test set.

Table 5 summarizes the performance of our proposed system (ArabicRAG) and the baseline (GPT-4o). ArabicRAG demonstrates superior performance across all the metrics. With a 60% higher rate of fully correct answers (score=3 as judged by the human experts), and 3 times fewer complete failures (score=0) compared to GPT. We can notice a smaller standard deviation in ArabicRAG (0.8) compared to GPT of 1.2. This means that ArabicRAG tends to have more predictable performance around the mean score. GPT in 15% of the test

[11] https://core.telegram.org/

Model	PyPDF OCR		Surya OCR	
	NDCG@5	P@5	NDCG@5	P@5
Okapi BM25	67.30	37.97	**73.15**	**45.50**
LMDirichlet	58.99	38.82	61.60	41.39
LM-JelinekMercer	65.84	41.55	69.17	43.10

Table 4: Performance of three retrieval models against the test dataset, both NDCG@10 and P@10 are reported in percentages, bold numbers are statistically significants

Metric	ArabicRAG	Baseline
Mean Score ($\pm SD$)	2.4 ($\pm$0.8)	1.5 ($\pm$1.2)
%Score =3	60%	40%
%Score =0	5%	15%

Table 5: Evaluation of the two bots on the test set. We report average of relevance graded score, and average of citation graded score for each system

set (around 10 questions) fails to provide a correct answer with accurate citations. For example, when GPT is asked to provide a brief introduction about historical figures from the Arabic literature, it tends to provide basic information as usually found in Wikipedia, but the main issue is with almost unrelated citations to books and articles. When we checked those references, we found out that they are irrelevant. This is considered hallucination. The pattern of synthetic scholarships poses particular risks in academic applications where source authenticity is crucial to the credibility of the research.

However, upon examining the failures in the ArabicRAG system, we observed that the inability to generate correct answers is primarily due to the limited number of research documents in the corpus (approximately 10,000). Consequently, the Information Retrieval system often retrieves documents that are poorly relevant to the questions, leading to irrelevant answers generated by the LLM. One solution is to enable real-time internet searches from reliable Arabic sources when the retrieved documents have low similarity scores. Another potential solution is to continuously expand the corpus by indexing more documents, thereby covering a diverse set of research topics. This is a potential future work of this research.

5 Conclusion

In this paper, we explored the impact of advanced Arabic language preprocessing techniques on the performance of information retrieval systems and their downstream influence on retrieval-augmented generation systems. Our findings suggest that employing a state-of-the-art deep learning-based OCR system (Surya) significantly enhances the IR performance, with improvements of up to 8 points in P@5 and 11% in RAG answering accuracy compared to baseline system. These results underscore the importance of robust preprocessing and language-aware IR in addressing challenges posed by Arabic script complexity and domain-specific terminology.

By integrating our enhanced IR system with a generative model, we developed ArabicRAG, a chatbot capable of providing contextually accurate and citation-rich answers to academic queries. Comparative evaluations against a baseline system revealed that ArabicRAG achieves a 20% higher rate of fully correct answers and significantly reduces hallucinations.

Future work will focus on expanding the corpus to cover a broader range of research topics and exploring real-time internet-based retrieval to address low-similarity cases. These enhancements aim to further improve the system's ability to deliver accurate and relevant responses, thereby advancing the state of Arabic NLP in academic contexts.

References

Gagan Bhatia, El Moatez Billah Nagoudi, Fakhraddin Alwajih, and Muhammad Abdul-Mageed. 2024. Qalam: A multimodal LLM for Arabic optical character and handwriting recognition. In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 210–224, Bangkok, Thailand. Association for Computational Linguistics.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

H. Bunke, M. Roth, and E.G. Schukat-Talamazzini. 1995. Off-line cursive handwriting recognition using hidden markov models. *Pattern Recognition*, 28(9):1399–1413.

Fabio Crestani, Mounia Lalmas, Cornelis J. Van Rijsbergen, and Iain Campbell. 1998. "is this document relevant?…probably": a survey of probabilistic models in information retrieval. *ACM Comput. Surv.*, 30(4):528–552.

Dima El Zein and Célia da Costa Pereira. 2022. User's knowledge and information needs in information retrieval evaluation. In *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '22, page 170–178, New York, NY, USA. Association for Computing Machinery.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, et al. 2021. Neurips 2020 efficientqa competition: Systems, analyses and lessons learned. In *NeurIPS 2020 Competition and Demonstration Track*, pages 86–111. PMLR.

Fabio Petroni, Federico Siciliano, Fabrizio Silvestri, and Giovanni Trappolini. 2024. IR-RAG @ SIGIR24: Information Retrieval's Role in RAG Systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 3036–3039, New York, NY, USA. ACM.

Jawad Sadek, Fairouz Chakkour, and Farid Meziane. 2012. Arabic Rhetorical Relations Extraction For Answering "Why" and "How to" Questions. In *Proceedings of the 17th International Conference on Applications of Natural Language Processing and Information Systems*, NLDB'12, page 385–390, Berlin, Heidelberg. Springer.

Simone Teufel. 2007. *An Overview of Evaluation Methods in TREC Ad Hoc Information Retrieval and TREC Question Answering*, pages 163–186. Springer Netherlands, Dordrecht.

Chengxiang Zhai and John Lafferty. 2017. A study of smoothing methods for language models applied to ad hoc information retrieval. In *ACM SIGIR Forum*, volume 51, pages 268–276. ACM New York, NY, USA.