Arabic Question Answering Systems Architectures: A Prisma-Based Systematic Literature Review

Kaouthar Omari Alaoui, Najima Daoudi and Manar Abourezq

Abstract— **The domain of Natural Language Processing (NLP) has undergone significant growth in the past few years, particularly in Arabic, a Semitic language with unique characteristics. Question answering systems (QAS) play a pivotal role in transforming raw data into actionable knowledge but designing effective QAS is a multifaceted challenge due to ambiguity in queries, complex sentence structures, and variations in question types. The importance of accurate and efficient Arabic QAS transcends theoretical realms, finding practical applications across various domains, including virtual assistants, search engines, and information retrieval (IR) systems. There is currently no universally standardized approach to developing QAS, but they have emerged as a viable solution for search engines. This study examines the current landscape of Arabic QAS, focusing on the overall architecture of Arabic QAS and the diverse sub-modules and techniques utilized in existing studies. It aims to highlight key advancements and the potential of future research to enhance the performance of Arabic QAS across different applications. In conducting a systematic literature review (SLR), an analysis was performed on 40 publications sourced from various databases to present an extensive examination of Arabic QAS. The choice of SLR methodologies was deliberate, as it guarantees the reproducibility of findings, promoting impartiality and clarity.**

***Index Terms*— Arabic-Natural-Language-Processing, Arabic-Question-Answering-System, Question-Analysis, Information-Retrieval, Answer-Extraction.**

I. Introduction

Natural language processing (NLP) has grown at an unparalleled rate in the last few years, catalyzed by advancements in machine learning and artificial intelligence. Among the plethora of languages under investigation, Arabic, with its rich linguistic heritage and unique characteristics, has emerged as a focal point for research and development. This work examines the complexities of Arabic Natural Language Processing (ANLP) in detail in this work.

Specifically, we will look at the difficulties and developments in the field of Arabic question answering systems (QAS). We will present the overall structure of a QAS in Arabic and the different sub-modules and techniques used in previous works to date.

Arabic, as a Semitic language, possesses distinctive features that set it apart from others. Its complex morphology, including root-based word formation and intricate syntactic structures, necessitates specialized approaches for effective processing. Additionally, the prevalence of multiple dialects poses a challenge in developing universally applicable language models. The journey towards robust ANLP systems is fraught with challenges. Limited labeled datasets, morphological complexity, and a scarcity of linguistic resources have hampered progress. Researchers also grapple with the task of accommodating the diversity of Arabic dialects, each presenting unique linguistic characteristics [1]. ANLP encompasses an array of tasks, each demanding nuanced solutions for the Arabic language. Named entity recognition (NER), sentiment analysis, part-of-speech (POS) tagging, summarization, machine translation, and question answering (QA) are among the pivotal tasks explored in the article. Understanding and addressing the intricacies of these tasks in the context of Arabic are crucial for the development of comprehensive language models.

In the 1990s, the emergence and widespread public accessibility of the Internet necessitated the development of tools for information retrieval (IR). This need arose from the challenge of determining the specific information that users were seeking, which has been a major focus in the field of IR. As a result, there has been an increased interest in the development of QAS. These systems aim to teach machines how to automatically respond to queries in natural language, spanning a broad spectrum of subjects and domains.

In effect, QAS play a pivotal role in transforming raw data into actionable knowledge. In the Arabic linguistic landscape, designing effective QAS is a multifaceted challenge. Ambiguity in queries, complex sentence structures, and variations in question types are obstacles that demand innovative solutions. The importance of accurate and efficient Arabic QAS transcends theoretical realms, finding practical applications across various domains. Virtual assistants, search engines, and IR systems can greatly benefit from improved QAS. In fact, QA is an evaluative task that holds substantial implications for users in various domains. It is a highly complex and tricky task that can be effectively utilized to evaluate the prowess of machine comprehension of texts, which has become a crucial aspect of NLP research. The overarching goal of

Manuscript received May 23, 2024; revised May 23, 2025.

Kaouthar Omari Alaoui is a Ph.D. student at the School of Information Sciences, Avenue Ibn Sina, B.P. 765, Agdal, Rabat, Morocco (e-mail: kaouthar.omari-alaoui@esi.ac.ma).

Najima Daoudi is a Full Professor at the School of Information Sciences, Rabat, Morocco (e-mail: ndaoudi@esi.ac.ma).

Manar Abourezq is a Professor at the School of Information Sciences, Rabat, Morocco (e-mail: mabourezq@esi.ac.ma).

QAS is to provide a coherent and accurate answer to a given question, which can be either extracted or generated from unstructured or structured text sources. These systems encompass a wide range of applications, including dialog systems, generating question systems, and community QAS, each designed with a specific purpose in mind. While a general QAS strives to enable machines to respond to diverse questions, other systems concentrate on different objectives. For instance, a community QAS primarily emphasizes IR rather than extracting responses. This kind of system is typically built by collecting a vast array of questions and corresponding answers from sources such as online forums and Q&A websites, thereby creating a comprehensive community QA dataset [2]. On the other hand, a generating question system operates inversely to a traditional QAS. It generates questions based on given passages by effectively leveraging knowledge bases [3]. Lastly, a conversational system aims to generate relevant responses for any type of text input [4], thereby facilitating seamless communication between humans and machines [5]. QAS are computer science applications that strive to facilitate the provision of accurate and coherent responses to inquiries made by people. They can be aptly considered as an extension of conventional search engines, with the main distinction being that search engines typically yield a set of pertinent documents as a result, thus leaving the task of Answer Extraction (AE) to the users themselves. In contrast, QAS deliver a single, concise answer, saving users valuable time and effort in navigating through an array of documents. The field of QA intersects with various domains within computer science, most notably NLP, human-computer interaction, IR, and artificial intelligence, thereby fostering interdisciplinary collaboration and advancements in these areas. QAS operate in close collaboration with IR and utilize techniques from both IR and NLP. They are a complex form of IR as they involve expressing information needs through natural language statements or questions, making them a natural form of human-computer communication. According to this system, a question is a natural language phrase that expresses the user's need for particular information and usually starts with an interrogative word.

There is currently no universally standardized approach to developing QAS, as architectural designs can vary greatly. However, QAS have emerged as a viable solution for search engines, as they aim to provide precise and accurate information. It is worth noting that there is a common structure or typical architecture observed in QAS, which generally consists of three distinct steps [6]. As a result, various approaches and methodologies have been proposed for the development of QAS in Arabic.

In this paper, we have retrieved a total of 40 publications from various databases, including Scopus, Science Direct, IEEE Xplore and ACM Digital Library, to provide a comprehensive overview of existing Arabic QAS. Through the utilization of a systematic literature review (SLR) approach, we have analyzed these articles to gain insights into the process, different components, algorithms, techniques, and tools employed in these systems. The adoption of SLR studies ensures the reproducibility of findings, thereby promoting objectivity and transparency. The introductory section of this paper offers a comprehensive overview of the topic at hand. Subsequently, the second section delves into the methodology employed in this investigation, encompassing the justification for running an SLR on Arabic QAS, the research objectives defined, the search strategy used, the screening process carried out to choose articles, and the technique applied to evaluate the quality of the selected studies. Proceeding to the third section, the SLR outcomes are exhibited following the predetermined research questions (RQs). Subsequently, the fourth section is dedicated to the discussion of the paper, while the fifth section concludes the SLR and suggests possible directions for further study.

II. MATERIALS AND METHODS

A. *Objective*

The primary objective of this paper is to review the existing literature on Arabic QAS, with the ultimate goal of exploring the various architectures and techniques employed across the three main components found in nearly all reviewed systems, which are: Question Analysis, IR, and AE. One purpose also of this study is to identify the most efficient techniques in the question answering NLP task on the Arabic language and the optimal sub-modules that can be added to improve any Arabic QAS.

B. *Methodology*

This systematic review was conducted using the PRISMA (Preferred Reporting Items for Systematic reviews and Meta Analyses) methodology [7], an established framework developed by Liberati et al. The review was conducted over a period spanning from April 2023 to February 2024.

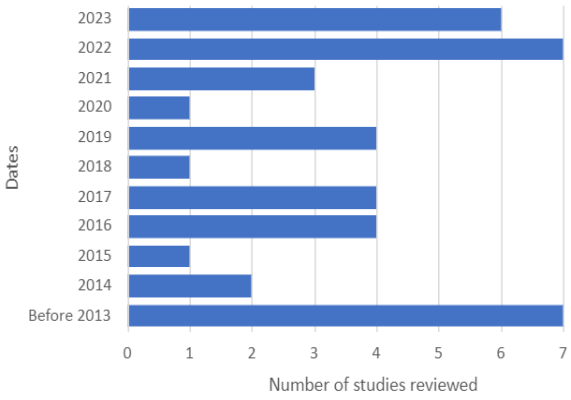

Fig. 1. Distribution of publication over the years 2002–2023.

While conducting this systematic literature review (SLR) on Arabic QAS, we used the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines [7], as depicted in Figure 2. This set of guidelines initially establishes the RQ. Subsequently, we performed an initial study and developed a search strategy including relevant terms and databases. The retrieved papers were subjected to a rigorous selection procedure that included exclusions based on the abstract, title, and keywords. We then carried out a selection process using inclusion and exclusion criteria. During the validation phase, we confirmed the validity of the collected papers and the reproducibility of the results. This yielded more than 40

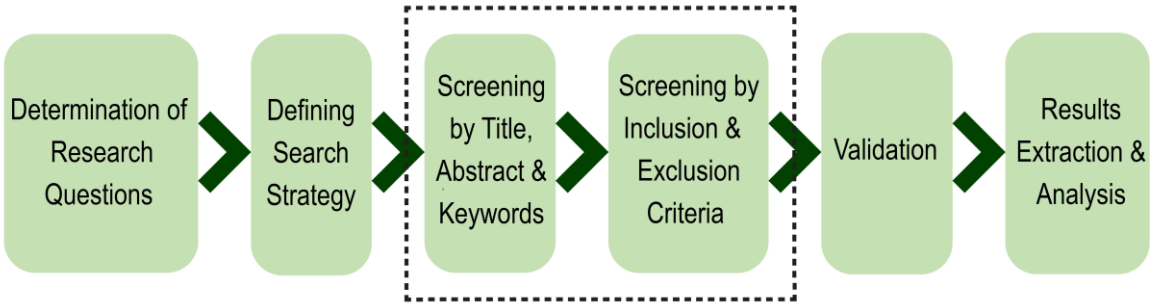

Fig. 2. The review protocol.

references spanning the period from 2002 to 2023 (Figure 1), providing an overview of the existing Arabic QAS, their modules, and the techniques, algorithms, and tools they employ. In the final stage of the review process, data was extracted from the documents in order to tackle the RQ. The Arabic QAS reviewed are diverse and can be classified into several typologies. For instance, we can cite: open-domain QAS, domain-specific QAS, community-based QAS, dialogue systems, keyword-based retrieval systems, IR-based systems, machine learning-based systems, knowledge base (KB) systems, rule-based systems, and semantic parsing systems. In the reviewed works, the three main common components existing in almost all the different QAS are Question Analysis, Information Retrieval (IR) or Document Retrieval (DR) or Passage Retrieval (PR), and AE. We conducted our analysis by examining each of these components and their respective sub-modules.

C. Research questions

RQs are clearly defined in accordance with the PRISMA guidelines. As previously mentioned, the three core components found in almost all Arabic QAS in the reviewed studies are: Question Analysis, IR, and AE. In this study, the RQs are as follows:

→ RQ1: What are the specific functions of each of the three steps in the QA process?
→ RQ2: What techniques are used in the Question Analysis component?
→ RQ3: What methods are employed in the IR, DR, or PR module?
→ RQ4: What techniques are applied in the AE component?

D. *Search strategy*

We designed search phrases that could effectively address the RQs. We adopted a comprehensive approach by querying five prominent academic databases: ACM Digital Library, ResearchGate, ScienceDirect, Scopus, and IEEE Xplore. We searched for “Arabic Question Answering Systems” from 2002 to 2023. These databases are widely recognized as reputable sources of high-quality publications in the fields of computer science and engineering. By gradually expanding the search queries using various related concepts, we were able to define our final search phrases.

Initial search terms included: "Arabic AND Question Answering Systems", "ANLP AND QAS", and "Arabic Question Answering Systems OR Arabic Chatbots OR Arabic Conversational Systems". Keywords and trends from some of the collected articles were then examined.

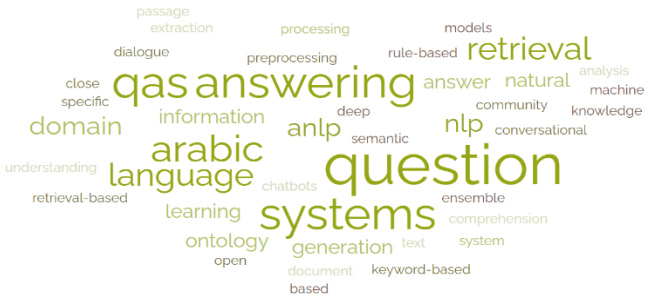

Fig. 3. Word cloud representing the most used keywords.

E. Article Selection by Screening

In accordance with the PRISMA 2020 guidelines [7], we reviewed the papers retrieved from the selected databases. Figure 5 presents a flowchart and an explanation of the selection procedure [8]. We obtained 654 articles, with 301 from Scopus, 133 from IEEE Xplore, 94 from ScienceDirect, 100 from ResearchGate, and 26 from ACM Digital Library.

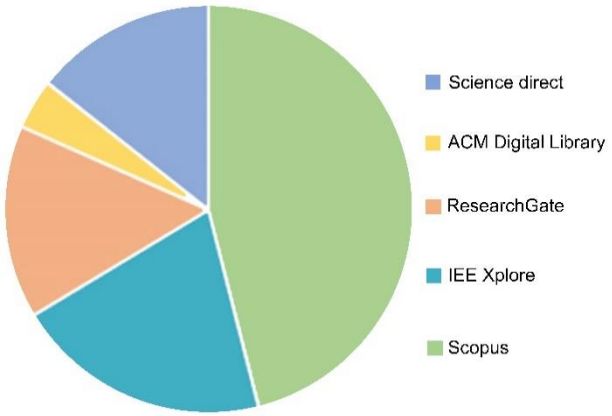

Fig. 4. Distribution of databases used to find relevant studies.

- **Screening by Title, Abstract, and Keywords**

After eliminating 16 duplicate entries, 638 papers remained. Of these, 540 were excluded based on their titles, abstracts, and keywords. We then examined the full texts of the remaining 98 publications.

- **Screening by Inclusive and Exclusive Criteria**

Eligibility criteria play a crucial part in the evaluation and determination of the soundness, comprehensiveness, and relevance of a review. These criteria are indispensable in the process of scrutinizing the adequacy and appropriateness of a review, as they enable researchers and reviewers to assess the suitability of the included studies. By establishing specific requirements and standards, eligibility criteria facilitate the identification of relevant studies and add to the total quality and reliability of the review. Therefore, the incorporation of well-defined eligibility criteria is crucial for ensuring the credibility and thoroughness of a review, while also enhancing its generalizability and applicability to the wider context. Thus, it is imperative to carefully formulate and implement eligibility criteria to preserve the rigor and authenticity of the review process. It is noteworthy to acknowledge that the exclusion criteria were formulated with a progressive approach. For instance, if an article fails to meet the inclusion criteria, it is promptly eliminated without any subsequent assessment of supplementary excluding standards.

Inclusion criteria (IC)

- ✓ IC-1 The paper is a research Article or Proceedings / Conference Article;
- ✓ IC-2 The title, abstract, or keywords must contain one of the following or equivalent phrases: "Arabic Question Answering Systems", "Arabic Chatbots", or "Arabic Conversational Systems".

Exclusion Criteria (EC)

- × EC-1 The study is not written in English;
- × EC-2 The full text of the study is not available;
- × EC-3 The work is not conducted on the Arabic language;
- × EC-4 The paper does not focus on the question-answering task;
- × EC-5 The study does not include any of the three core components of a QAS mentioned previously.

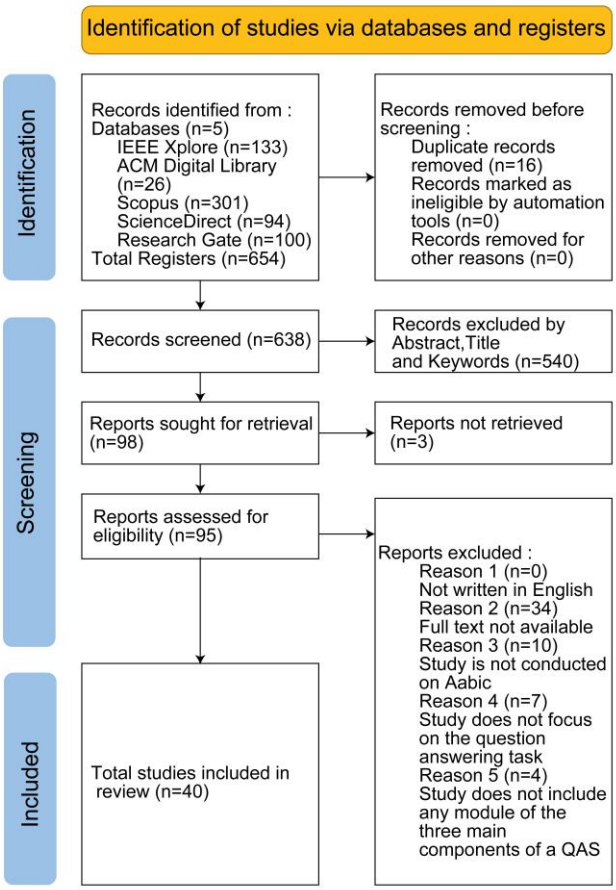

Fig. 5. PRISMA 2020 flow diagram of the study selection.

F. Validation

All relevant studies were successfully retrieved. The search results from each database were exported in BibTeX format, and the selection process was carried out using the Zotero reference management tool. Each article was carefully examined to assess its quality, ensuring the reproducibility of findings with objectivity and transparency.

III. Results

A. Databases

We begin presenting the results of our SLR by listing the datasets used for training and testing in the reviewed studies. Most of these datasets are publicly available.

TABLE I
DATASETS USED IN THE REVIEWED STUDIES .

Study	Dataset
[8]	Al-Raya newspaper content.
[9]	Linguistic development environment.
[11]	Web documents.
[12]	Not specified.
[13]	A corpus of 20 Arabic documents.
[10]	Web corpus.
[15]	Arabic corpus of 39 660 words.
[16]	Holy Quran verses and interpretation books.
[17]	UIUC, 500 records for training and 500 for testing.
[18]	Research papers.
[19]	Web documents.
[14]	Web documents.
[20]	A collection of fatwas from Ibn-Othaimeen prayer fatawas book.
[21]	Web documents.
[22]	Web corpus.
[23]	Web documents.
[24]	Web corpus.
[26]	100 questions in the pathology ontology domain.
[27]	TREC and CLEF.
[28]	Wikipedia.
[29]	-Wikinews (Darwish & Mubarak, 2016); -SPMRL 2013 (Seddah et al., 2013); -Online forum content.
[6]	10,000 documents in 10 classes.
[30]	Web corpus.
[31]	-SQuAD; -CQA-subtask D.
[32]	100 questions.
[1]	-DAWQAS; -Bakari et al. factoid dataset; -ARCD.
[33]	Web documents.
[34]	SOQAL.
[35]	Web documents.
[36]	Web corpus of questions and texts.
[39]	-Popular Yahoo!; -Answers community platform;
[37]	QRCD.
[40]	Student queries from social media and forms.
[38]	Not specified.
[41]	Fusion of ARCD, Arabic SQuAD, XQuAD, and TyDi QA.
[42]	-ARCD; -TyDiQA.
[43]	Arabic Wikipedia.
[47]	-SQuAD; -CLEF and TREC; -Arabic news websites; -Cite-corpus; -OSIAN; -Tweets; -Arabic Wikipedia; -Mawdoo3 articles.
[44]	5000 images from MS-COCO.
[45]	Educational websites.
[19]	Web documents.
[46]	-Arabic-SQuAD; -ARCD.

B. General architecture of an Arabic question-answering system

In this part, we will answer RQ1: What are the specific functions of each of the three steps in the QA process? As we mentioned before, all the Arabic QAS reviewed consist of three main components: question analysis, document, passage, or IE, and AE [1].

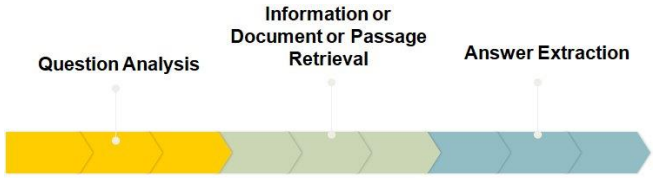

Fig. 6. General QAS process.

The first component of QAS is question analysis, which is responsible for analyzing a given question. During this initial stage, the question is meticulously examined in terms of its semantics and syntax, with the aim of extracting the user's intention, highlighting the pivotal keywords, and generating the inquiry. This phase of the question elucidates its focal point or primary objective. This step typically involves several subtasks such as question classification, question segmentation, keywords extraction, question formation, answer type detection, and query expansion. Question segmentation refers to breaking down the question into smaller parts or segments to better understand its structure and meaning. Question classification involves categorizing the question into different types based on a pre-established categorization and the anticipated type of response. Question formation focuses on transforming the question into a more suitable format for processing and retrieval. Answer type detection aims to identify the kind of response anticipated by the question, such as a person's name, a date, or a location. Keywords extraction involves extracting important keywords or terms from the question to guide the retrieval process. Query expansion is the process of expanding the initial question with extra relevant words or synonyms to improve retrieval performance.

The second component of QAS is DR, which takes the processed query from the first component as input and aims to retrieve relevant documents. This step involves various tasks such as paragraph retrieval, short answer retrieval, ranking of relevant documents, sentence retrieval, identification of relevant documents, and PR. Sentence retrieval focuses on retrieving individual sentences or phrases from the records that most probably have the response to the query. Short answer retrieval aims to directly retrieve short answers to the question without additional processing. Paragraph retrieval involves retrieving entire

paragraphs that are pertinent to the inquiry. Identification of relevant documents involves determining which documents in a given collection are likely to contain the answer. Ranking of relevant documents includes ordering the retrieved documents based on their relevance to the question. PR focuses on retrieving specific passages or sections from the records that have the greatest chance of having the correct answer.

The last component of QAS is AE, which obtains the pertinent documents from the second component and the processed question from the first component. Its objective is to select from the retrieved documents the most pertinent response to the query. AE may involve various tasks such as various NLP methods, such as response validation, answer scoring and rating, answer selection, answer display, and name entity recognition (NER). The goal of the NLP subtask known as "NER" is to locate and classify named entities in text, such as names of individuals, groups, places, and dates. Answer validation involves verifying the correctness or validity of the extracted answer. Answer scoring and rating involve assigning scores or ratings to different candidate answers based on their quality or relevance. Answer selection includes choosing the best answer among the candidate answers based on certain criteria. Answer presentation focuses on formatting and presenting the final answer suitably for the user.

In conclusion, QAS comprise three main components: question analysis, DR, and AE (Figure 6). Question analysis involves analyzing the given question through various subtasks. DR intends to retrieve relevant documents based on the processed question. AE focuses on extracting the most relevant answer from the retrieved documents using different NLP techniques. These components work together to enable QAS to respond to customer inquiries in a precise and instructive manner.

C. Standard Phases of Arabic Question Answering Systems and Techniques Used

- **First component: Question analysis**

Through the examination of all QAS proposed in the studies reviewed, in this section we will answer the second research question RQ2: What techniques are used in the Question Analysis component?

In 2002, Hammo et al. [8] proposed a QAS referred to as the QARAB. The system starts with the preprocessing step. It considers the incoming query as a "bag of words" that is used to search the index file and retrieve a ranked collection of documents that may have the answer. The process of dealing with the query starts by extracting individual terms through tokenization. Subsequently, the stop-words are eliminated. The remaining words are assigned POS tags to emphasize the keywords that should be present in the postulated answer. Moreover, the procedure involves also treating each individual document (specifically, paragraphs) in a similar manner to how the question is processed, thereby retrieving sentences that potentially encompass the answer. Then, QARAB processes query expansion. In order to enhance the effectiveness of search and retrieval outcomes, the query is extended to encompass all the terms (both verbs and nouns derived from verbs) that are present in the index file and share the same roots as the original query words that were extracted. Subsequently, the processed query outcome is transmitted to the IE system in order to obtain a prioritized compilation of documents that correspond to the terms within the query. Before moving to query keyword identification, the query type is determined. Based on a list of recognized "question types," questions are categorized. Based on a list of recognized "question types," questions are categorized. These different question types assist the authors in figuring out what kind of processing is required to locate and retrieve the final answer. Lastly, POS tags are applied to the remaining words in the query. The Type-Finder & Proper Name-Finder system developed by Abuleil [1999] is used in this procedure. Verbs are easiest to identify due to clear morphological patterns. Nouns, especially proper nouns, help find the expected answer from relevant documents. They must occur in the same order as in the question and in the chosen answer section. To help in the identification of proper names, a collection of Arabic keywords for personal names, organization names, localities, numbers, money, and dates has been prepared.

For QASAL [9] proposed in 2009, the question analysis module is designed to handle any factoid question in Arabic. Its purpose is to establish the best answer by gathering as much pertinent information as possible from the question. This includes identifying the expected answer type, which is used by the AE module, as well as determining the question focus, which involves identifying named entities that may be important for finding potential answers. Additionally, the module provides a list of crucial keywords that can be used by the PR module to perform a search query. The authors use NooJ's linguistic engine.

In 2010, DefArabicQA [10] was developed, where the question analysis module is crucial since it also identifies the main query and ascertains the anticipated type of response. Lexical question patterns are used to determine the question topic, and the interrogative pronoun is used to infer the type of the expected response. In the case of QArabPro (2011) [11], the system processes a question analysis-query reformulation. It views the incoming question as a collection of words. The authors apply stemming and NLP techniques to assign each word its root and POS, storing them in a database. The NLP system includes modules for tokenizing, tagging, feature-finding, and NER. Additionally, they perform query expansion and query type identification. In IDRAAQ (2012) [12], the process of analyzing a question involves the extraction of its keywords, the identification of the anticipated answer's structure, and the formulation of the query that will be subsequently transferred to the PR module.

Regarding the Yes/No Arabic QAS [13] proposed in 2013, the task of the question analysis module includes removing the question mark and interrogative particle, tokenization, removing stop words, and removing negation particles if they exist. Additionally, the module sets the

negation property of the question representation. In the tagging procedure, authors employ a tagger to identify the sort of word (verb or noun) and identify its root. Lastly, they parse the Arabic sentence after the interrogative particle, which can be nominal or verbal. Whereas, always in the same year, Fareed et al. [14] proposed a semantic Arabic QAS based on Khoja Stemmer and AWN (Arabic Word Net). In the first phase, the question type and queries are identified. They used AWN ontology and DB tables to identify relations. These relations expand the question and provide alternative queries. A Java program is used for processing.

Another Arabic QAS is JAWEB (2014) [15] which includes a question-analyzer. The purpose of this component is to determine the question's type. It consists of five modules: a question keyword extractor, a tokenizer, an answer-type detector, a question word stemmer, and an extra keywords generator. In the same year, in Al-Bayan [16], the Arabic question is preprocessed to extract the query for IE. The question is also classified to determine the type of question and its expected answer for AE. Preprocessing uses MADA for stemming, lemmatization, POS tagging, glossing, disambiguation, and diacritization. Support vector machine (SVM) classifier and a new taxonomy are used for question classification.

In 2015, Chalabi [17] considered that the question analysis phase has two subtasks: Classifying questions and Expanding queries with additional keywords. The authors utilized tools such as Nooj and AWN to complete these tasks. Then, in 2016, Ahmed and Anto [18] split the question into tokens for question tokenization, and remove prepositions, conjunctions, and interrogative words for stop words removal. They also expand the question using semantic IE techniques. They extract classes using a trained SVM classifier and identify the focus for ranking candidate answers. The authors extract the focus chunk by applying a rule-based method. Based on POS Tagging information, rule-based chunking is used for nouns and noun phrases. The Stanford parser is used for parsing Arabic questions. In the same year, Bakari et al. [19] affirmed that question analysis involves constructing a question representation, classifying user questions, deriving expected answer types, extracting keywords, and reformulating a question into semantically equivalent multiple questions. On the other hand, always in 2016, Shakir and Sheker [20] proposed a Domain-specific ontology-based QAS where the preprocessing phase seeks to examine the user's typed question by removing irrelevant data like numbers, stop-words, and punctuation. Using similarity metrics like Cosine and Jaccard, the Question Analysis step seeks to identify questions that are similar within the dataset. Furthermore, the authors process a query expansion to provide semantic correspondence utilizing the open-domain ontology in order to expand the typed inquiry. The proposed knowledge source for the Islamic Fatwa domain is the specific-domain ontology, which has been constructed using Term Frequency - Inverse Document Frequency (TF-IDF) from the dataset. The Open-domain Ontology is AWN, which contains large-scale semantics in Arabic including relations. Furthermore, in the work of Ray and Shaalan (2016) [21], the process starts with preprocessing performing encoding and normalization, then question analysis analyzes the user's typed question in terms of morphology, aiming to identify the most similar question from the dataset for the exact answer. Two similarity measures are used Cosine and Jaccard. Furthermore, query expansion reformulates the user's query to retrieve relevant documents. Query words are analyzed for semantics and morphology, as well as spelling errors. A specific-domain ontology for Islamic Fatwa on Islamic praying was constructed in this study. The main classes of ontology were created using TF-IDF from the dataset. The study utilized an open-domain ontology known as AWN.

In 2017, an Intelligent QAS proposed by Ahmed et al. [22], the question processing module includes three operations: question classification, question focus detection, and temporal inference. The analysis of questions to determine the distinct question classes and anticipated answer type based on named entities is the goal of question classification. A question's focus, on the other hand, is a noun phrase that indicates and makes clear the kind of response that is anticipated. The question's main noun is usually the focus. Additional NLP preprocessing steps including stop word removal, tokenization, stemming, and question expansion are conducted. Furthermore, temporal signals in the question are forwarded for further processing.

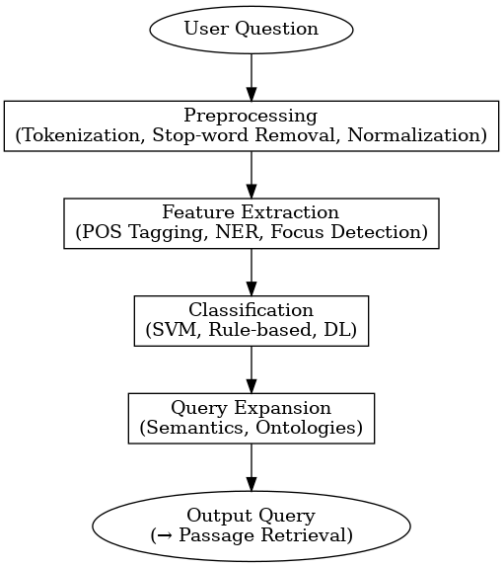

Fig. 7. Question analysis pipeline.

LEMAZA [23] was also proposed in 2017, where the primary function of the question analysis component is to preprocess the question and extract keywords to represent the user's need. It relies on NLP tools for linguistic analysis, and in order to obtain ranked documents, it uses the why question as a "bag of words". The authors apply stop-word removal, normalization, tokenization, and stemming algorithm to obtain roots. The system formulates, generates the query, and expands it with synonyms and words sharing the same root. We find also AlQuAnS (2017) [24], where the authors apply preprocessing operations including normalization, stemming, POS tagging, and stop words removal using MADAMIRA [25]. Then they utilize the AWN ontology and the Super Upper Merged Ontology (SUMO) for query expansion. They define four sub-processes for query expansion based on semantic relations

in AWN. The SVM classifier is used for Question Classification, as it has demonstrated the best outcomes from their tests. Question classification requires taxonomy, which is based on the work of Li and Roth (2002). They build a classifier model to test input questions against this taxonomy. Albarghothi et al. [26] also proposed an ontology-based system in 2017. In this work, the authors built a QAS based on ontology. Before the question analysis step, they start by constructing the dataset with ontology and the semantic web by employing the Protégé tool, yielding RDF/OWL file. Using information from ontologies, questions are selected, and keywords are generated from the questions for SPARQL query parsing. Then, the questions are processed to generate keywords for the SPARQL query. The questions are analyzed based on NLP tasks. Keywords from questions are needed for ontology mapping. Question processing steps include tokenization, stop-word removal, normalization, and POS tagging. POS tagging assigns question keywords to SPARQL queries to match with ontology objects and data types, and the keywords generated from questions are matched with SPARQL.

Later, in 2018, Lahbari et al. [27] preprocess questions through tokenization and a morphological analysis, then they adopted a machine learning approach to categorize questions using SVM classifier. Taking into account the semantic interpretation of the answer type, they decided to implement the taxonomy proposed by Li and Roth. They expand queries using AWN and a hybrid Arabic POS tagging scheme.

Afterward, in 2019, four Arabic QAS were proposed. The first system, SOQAL [28], where the proposed reader is BERT, a language model that is currently leading in the SQuAD leaderboard. BERT's core model is a bi-directional Transformer. The input text is tokenized and embedded using a shared vocabulary, with Arabic diacritics removed. Each input point is represented as a single sentence. Then, Romeo et al. [29] proposed an Arabic community QAS. For Arabic processing, they utilized their Farasa tool. The pipeline is built on UIMA and it consists of a named entity recognizer, a segmenter, a dependency parser, a constituency parser, a diacritizer, and a POS tagger. Moreover, we have AQAS [6], where the authors start with building a SVM Model for classifying the queries. They begin with preprocessing where the documents are tokenized into words and stop words are eliminated using an Arabic stop words list. A stemmer is used to eliminate word suffixes and minimize the number of words. Then, they move to feature extraction and create a feature vector for each document using distinct terms. The values are calculated using TF-IDF, indicating the importance of a word in a set of documents. Finally, they build the SVM Model which is a classifier that separates instances using a hyperplane. It determines an optimal line to separate categories in supervised learning. Last, we find EWAQ [30], where the preprocessing phase begins with question analysis, which involves removing stop words. Abu-Elkhair's stop word list is used in the system. Next, word stemming is performed. Finally, for every word in the question and the passages they have found, the authors extract related terms. EWAQ increases the accuracy of IE systems by extracting all senses that can be associated with each word in the question and passages using AWN. Later, in 2020, Almiman et al. [31] proposed a community question-answering approach where the preprocessing step involves using the SPLIT technique to obtain tokens. Dates and numbers are swapped out for the tokens 'Num' and 'Date' respectively, stemming and lemmatization are performed using MADAMIRA [25], and stop words are also eliminated.

In the next year, Maraoui et al. [32] proposed an Arabic factoid QAS for Islamic sciences using normalized corpora where the question processing step includes keywords extraction, NER and query formulation. Question analysis aims to formulate the query from the user question. The query contains information about the topic, named entities, and keywords. Preprocessing removes stop words and unnecessary particles and tokenizes the question topic using dictionaries. This is followed by lemmatizing and NER. Furthermore, matching the terms in the question to the Arabic language syntax is necessary. Patterns are designed to classify questions based on the terms used. Each pattern represents a list of questions regarding the same subject. Furthermore, Biltawi et al. [1] affirmed that question analysis techniques consist of question classification, domain classification, and NLP techniques such as NER, tokenization, segmentation, POS tagging, and parsing. Preprocessing techniques employed in Arabic QAS for questions and answers involve tokenization, noise removal, normalization, stemming, and rooting. And query expansion techniques, including the use of external resources like WordNet, can enhance IE performance through synonym extraction and inflectional-based expansion. Additionally, Hamza et al. [33] suggested a classification scheme for questions based on the continuous distributed representation of words and taxonomy. Preprocessing the text is where they begin. Next, they use two distinct approaches to construct question vectors: the novel model that improves word vectors using sub-word information, and TF-IDF weighting with n-gram. Every word is depicted by the authors as a sack of characters, and it is integrated into the collection of its n-gram. The total of a word's n-gram vectors is the word vector. Arabic questions are categorized by the categorization module using a certain taxonomy.

Afterwards, in 2022, Premasiri et al. [34] utilized transfer learning with transformers in QA through AraELECTRA-discriminator. Firstly, transformer models for the machine reading comprehension (MRC) task use a single sequence input with a question and paragraph separated by a [SEP] token. The model then adds a start vector and an end vector. Then, Faris et al. [35] suggest a deep learning approach for question classification. Dense feature representations and implicit relationship extraction are two capabilities of deep learning. The suggested model handles Arabic in healthcare by utilizing long-short-term memory (LSTM) and biLSTM networks.

In another study [36], the authors process a pre-treatment and question transformation. Pre-treatment involves analyzing questions beyond simple keyword segmentation

by eliminating stop words. They also expected answer types: place, person, date, organization, and numerical expression. The transformation involves generating reformulations of questions in declarative form by removing specific features and interrogation particles. And the eliminated information is deemed unimportant. Elkomy and Sarhan [37] used a Post-Processed Ensemble of BERT-based Models. And Aftab and Malik (2022) [38] proposed a model employing a pipeline architecture starting with data preparation. The Qur'anic Reading Comprehension Dataset (QRCD) training data must be processed to convert it into a structure suitable for input to BERT. The next step is to initialize BERT with either initial configurations or a generic pre-trained checkpoint. Another work conducted in 2022 is [39], where preprocessing is crucial for enhancing the cleanliness and processability of question collections. By removing unnecessary terms and filtering natural language community questions, the question preprocessing module attempts to convert them into a formal representation. Tasks including text cleaning, tokenization, stop word removal, and stemming are included in this module. Punctuation, non-letters, diacritical marks, and special characters are eliminated. English letters are converted to lowercase, while dates and numerical digits are normalized to specific tokens. In the case of Arabic question collections, orthographic normalization is also performed, consisting of letter normalization, stretching deletion, and diacriticals elimination.

Furthermore, in [40], an Arabic QAS was proposed for university students. The student initiates the query and activates the word-level completion, NER, and next-word prediction elements. The predictions are based on frequency and individual usage history. Each completion is processed using the NER model, and the predictions are generated for each completion. The predictions improve with more input words. Variants of queries are mapped to standard suggestions and classified based on entities. The standard queries are manually classified, and the suggestions may be re-ranked. The selected suggestion is converted to a structured query for retrieving the answer from the database. The selection click is saved for evaluation purposes.

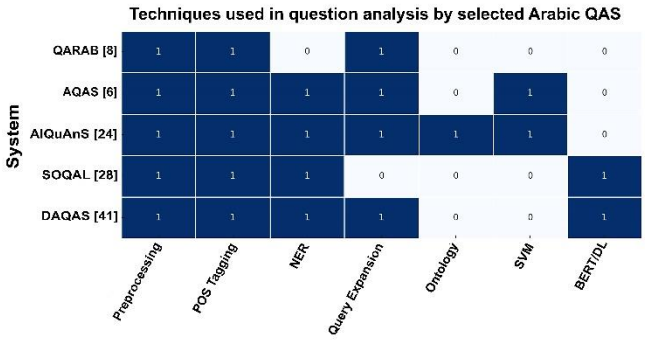

Fig. 8. Question analysis techniques heatmap.

Later, in DAQAS (2023) [41], the authors suggest a new dense duplicate question detection module that reduces response time for questions with stored answers and provides trusted exact answers. The module uses two dense encoders to create vectors from input and well-known queries. Similarity vectors are used to predict duplicate questions. After training, embeddings of previously answered questions are computed and indexed offline. The system retrieves the top candidates with the same response as the query provided. A BERT-based classifier determines the duplicate question with the highest score and returns the known answer. If no duplicate questions are detected, the question is passed to the retriever module.

In the same year, Alkhurayyif and Sait [42] proposed an Open Domain Arabic QAS using a deep learning technique. In the first step, keywords extraction requires procedures like orthographic ambiguity minimization, diacritization, tokenization, and morphological analysis. The authors use the Multinomial Naive Bayes (MNB) classifier to classify datasets and their relationships into different categories. Finally, to automate the process of answering questions, they combine Quantum LSTM (QLSTM) with ELMo vectorization. The authors take into account the orthographic and diacritical forms of the content. They consider the role of phonetic symbols from several perspectives. Therefore, diacritics are retained in complex terms to preserve semantic clarity. In order to eliminate orthographic ambiguity, the authors assess each term's function inside a phrase. Furthermore, automated preprocessing is used to eliminate unnecessary words. The Arabic dataset is classified by the authors using the MNB-NER classification technique. Text, question, topic, and context are the main categories. The context of a user's inquiry refers to the specific setting in which it occurs. The dataset is divided into four themes by the authors. Text and questions are grouped according to topic and context. Text is categorized into classes according to how likely it is to appear in another category. Terms that are related are grouped together through the labeling process. The authors created an Arabic QAS using QLSTM and ELMo. After converting user questions into vectors, the ELMo model looks for requests that are comparable in the query storage module. The matching procedure will assist the suggested QAS in providing a prompt response to the user's inquiry.

Also, Alruqi and Alzahrani [43] used large question and answer datasets and a corpus of textual documents for dataset preprocessing. They removed stop words, special characters, and noisy words. They also cleaned the corpus by removing irrelevant information. Then, Lahbari and El Alaou [47] affirmed that the purpose of the question analysis component is to develop and classify a user's query. The preprocessing stage eliminates foreign characters, diacritical markings, and punctuation. Stop words are not eliminated as they can be used as interrogative tools. After classifying and identifying query types, they eliminate them. The remaining portion is tokenized using white space. A lemmatizer is applied and POS tags are added to the text. Arabic lemmatizer assigns a distinct lemma to every word based on its context. Lemma is used as input for linguistic dictionaries. POS tagging provides information about syntactic category, gender, tenses, etc. Named entities are identified using a POS tagger, decreasing the number of passages to be retrieved and making the process more focused. POS tagger is also used to remove verbs from each question as nouns determine the meaning in Arabic sentences. Taxonomies are used to categorize information in a structured architecture. The machine learning approach is utilized to categorize questions using two taxonomies. SVM is a query classification technique. Upon completing preprocessing, lemmatizing, POS tagging, and classification, the queries are prepared for the subsequent step: retrieve documents and passages.

TABLE II
QUESTION ANALYSIS METHODS

Study	Techniques & Tools
[8]	Preprocessing: tokenization, stop words removal, POS tagger; Keywords extraction; Query expansion; Query type identification.
[9]	Expected answer type identification; Question focus determination (NER) / *NooJ's linguistic engine*.
[11]	Stemming, tokenizing, tagging, feature-finding, NER, POS; Query expansion; Query type identification.
[12]	Keywords extraction; Expected answer's structure identification; Query formulation.
[13]	Removing stop words, question marks, interrogative and negation particles, tokenization, tagging, and parsing.
[14]	Question type and query identification; Query expansion / *AWN ontology, DB tables, and Java program.*
[15]	Question's type determination by tokenizer, answer-type detector, question keyword extractor, extra keywords generator, and question word stemmer; Question classification.
[16]	Preprocessing: POS tagging, diacritization, lemmatization, disambiguation, stemming, and glossing / *MADA*; Query classification / *SVM classifier and a new taxonomy*; Expected answer identification.
[17]	Classifying questions and expanding queries / *Nooj and AWN.*
[18]	Tokenization, stop words removal, Parsing / *Stanford Parser*; Question expansion / *semantic IR techniques*; Class extraction / *a trained SVM classifier*; Focus identification / *a rule-based technique based on POS tagging.*
[19]	Constructing a question representation; Classifying user questions; Deriving expected answer types; Extracting keywords; Question reformulating.
[20]	Removing irrelevant data like numbers, stop words, and punctuation; Finding similar questions/ *similarity measures;* Query expansion / *AWN*.
[21]	Preprocessing: encoding and normalization; Similar questions identification (*Two similarity measures: Cosine and Jaccard*); Query expansion / *AWN.*
[22]	Preprocessing: stop words removal, tokenization, stemming; Question expansion; Question focus detection; Temporal inference; Question classification, and answer type expectation / *NER.*
[23]	Keywords extraction, tokenization, normalization, stop words removal, stemming, and Query expansion.
[24]	Preprocessing: normalization, stemming, POS tagging, and stop words removal / *MADAMIRA;* Query expansion / *AWN ontology and SUMO ontology;* Question classification / *(SVM) classifier & Li and Roth (2002) taxonomy*.
[26]	Question preprocessing: normalization, tokenization, stop words removal, and POS tagging.
[27]	Preprocessing: tokenization and a morphological analysis; question classification / *SVM classifier; S*emantic interpretation and answer type expectation / *the taxonomy put forth by Li and Roth;* POS tagging; Query expansion / *AWN*.
[28]	Tokenizing and embedding text / *a shared vocabulary;* Removing Arabic diacritics; Representing each input point as a single sentence / *BERT.*
[29]	Preprocessing: segmentation, POS tagging, NER, dependency parsing, constituency parsing, and diacritization /*Farasa*.
[6]	Preprocessing: tokenization, stop words elimination, stemming; Queries Classifying / *SVM model;* Feature extraction and creation of a feature vector for each document/ *TF-IDF*.
[30]	Preprocessing: removing stop words / *stop words list of Abu-Elkhair,* word stemming; Query expansion/ *AWN.*
[31]	Preprocessing: tokenization / *SPLIT technique,* stemming and lemmatization / *MADAMIRA,* stop words elimination.
[32]	Preprocessing: stop words removal, tokenization, lemmatizing and NER; Key words extraction; and Query formulation.
[1]	Preprocessing: tokenization, noise removal, normalization, stemming, and rooting; Question classification; Domain classification; Query expansion / *WordNet, synonym extraction and inflectional-based expansion.*
[33]	Question classification / *TF-IDF weighting with n-gram.*
[34]	AraELECTRA-discriminator with transfer learning.
[35]	Question classification / *LSTM and BiLSTM.*
[36]	Pre-treatment: keywords division, stop words elimination; Question transformation; Expected answer type identification: *place, person, date, organization, and numerical expression*.
[39]	Preprocessing: text cleaning, tokenization, stop words removal, and stemming, diacritics removal, stretching removal, and letter normalization.
[40]	Word-level completion; NER / *CAMeL Tools with Rule-Based NER & Machine Learning NER*; Next-word prediction elements / *LSTM*; Mapping variants to standard queries; Data preprocessing: Removing words duplicate, non-Arabic characters, diacritics, tide in letters and stop words, normalization; Question classification; Translating standard queries to structured query language.
[37]	*BERT*
[41]	Dense duplicate question detection / *two dense encoders, Similarity vectors & BERT-based classifier;* Preprocessing: removing stop words, diacritics, and kashidas, normalizing letters, segmenting and tokenizing / *Farasa segmenter*; Queries expansion / *mT5-small, AraGPT2-base, and AraGPT2-large.*
[42]	Preprocessing: diacritization, orthographic ambiguity minimization, tokenization, irrelevant words removal and morphological analysis; Keywords extraction; Query classification, Query conversion / *MNB classifier, QLSTM, ELMo model; MNB-NER classification technique, Similar query identification.*
[43]	Preprocessing: removing special characters, stop words, noisy and irrelevant words.
[47]	Preprocessing: tokenization, lemmatization, punctuation, diacritical markings, and foreign characters elimination; Query classification / *taxonomies & SVM*; NER / *POS tagger*.
[44]	Cleaning, normalization, tokenization, word embedding, merging, and padding.
[45]	Preprocessing: cleaning, tokenization, padding, stop words removal and stemming; Questions categorization.

Otherwise, in VAQA [44], a visual Arabic question answering system developed in 2023, the initial question is processed through multiple steps: cleaning, normalization, tokenization, word embedding, and merging and padding. Cleaning involves removing non-alphabetic and non-numeric symbols. Normalization reduces confusion by rendering letters into a single form. Tokenization splits the question into words, considering whether the question tool should be separated. Word embedding encodes words into numerical representations. Merging and padding trim questions to a fixed length and merge word embedding vectors. Finally, A. Alazzam et al. [45] proposed an Arabic chatbot, where the collected questions on education were used to create a dictionary. The questions were categorized using three intents: tag, pattern, and response. Each tag represents a specific topic and has associated patterns and responses. The dataset was processed using Python and

Natural Language Toolkit (NLTK) to clean the text, tokenize words, pad sequences, and create dictionaries. Tokenization splits the text into words, while padding ensures uniform sequence length. A dictionary was also created to map words to numerical equivalents and stop words were removed, in addition to stemming.

- **Second component: Information or Passage or Document Retrieval**

Through the examination of all QAS proposed in the studies reviewed, in this section we will answer RQ3: What methods are employed in the IR, DR, or PR module?

We start with QARAB [8], where the IR system is founded on Salton's vector space model and extracts responses to queries in natural language from the text collection of the Al-Raya newspaper. The IR system's objective is to look for relevant documents based on user queries. In the IR system, newspaper articles are saved in text format using a specific encoding scheme. The system's construction makes use of a relational database model and incorporates term weighting, tokenization, stop-word removal, and root extraction. For QASAL [9], the system retrieves relevant passages for the expected answer. Text passages serve as an essential conduit between documents and responses. They are a natural response unit for QAS. The system's effectiveness heavily relies on finding relevant passages. If no passage is found, the AE process will fail. In DefArabicQA [10], the PR module gathers the top-n snippets found that the search engine returned. The question topic determined by the question analysis module comprises the specific query. Only the snippets that contain the integrated question topic are retained after the top-n snippets have been gathered, based on heuristics, such as requiring snippet length to exceed 13 characters. In QArabPro [11], the constructed IR system has various database relations. The root table stores distinct roots of extracted terms. Root extraction is done with a stemmer. Information about documents is kept in the documents table. The verb table stores verbs of words. The stop word table contains stop words for the Arabic language. The variants table lists many words from papers that have the same format. Categoryized root information is contained in the document type root table. Furthermore, document processing is a crucial step for any IR technique. This step includes term weighting, root extraction, tokenization, and stop word removal. In IDRAAQ [12], the list of passages retrieved by the IR process is handled by the PR module. It ranks the passages to improve their relevance. The PR module of IDRAAQ has keyword-based and structure-based levels. The keyword-based level includes a semantic QA process. The IDRAAQ system worked on a new level called the semantic reasoning level. This level compares question and passage representations using Conceptual Graphs.

For what concerns Yes/No Arabic QAS [13], text processing and retrieval include two steps: First, splitting documents into paragraphs and retrieving the top five paragraphs using an indexing scheme. Second, ranking documents and retrieving the top five, then using an indexing scheme to retrieve the top five paragraphs. Whereas a semantic Arabic QAS based on Khoja Stemmer and AWN was proposed by Fareed et al. [14]. The PR module receives queries and generates top-ranked passages. It uses the Google search engine to find related snippets. The module considers the query structure and not just the presence of keywords in the passages. It also makes use of a PR system based on structure called JIRS to raise Google's ranking. To identify pertinent passages, the JIRS compares n-grams from the passage and the question using a model. For stemming, the authors use Khoja stemmer which is a system used to remove affixes from words and reduce them to their roots. In JAWEB [15] (2014), the IR component searches for relevant answers by matching keywords in information sources. Arabic NER will be used to provide more accurate answers. In Al-Bayan [16], a Quranic ontology is created to classify Quran verses based on their topics. Machine learning techniques are used to create a Semantic Interpreter, which maps natural language text to Quranic concepts. Cosine similarity between the input query and each verse in the Quran is computed to determine which verses score highest in relation to the user question. In another work [17], the DR module that receives the classified question relies on identifying the retrieval system's subset components, which comprise terms from the presumptive question. The retrieval system returns the most probable documents containing the answer. These documents are then analyzed by the "Document Analysis" sub-module. The document analysis module uses the question classification description to generate closely related answers. These answers are then sent to the "Answer Selection" module.

On the other hand, Ahmed and Anto [18] employ the VSM for the DR module. From the top 10 retrieved documents, the system extracts text. The top three documents are picked for further processing by the AE module. While Bakari et al. [19] affirmed that relevant information is retrieved from a knowledge base or document collection based on the user's query. Furthermore, in the work of Ray and Shaalan [21], document processing involves finding pertinent documents, prioritizing them, and retrieving the important passages. For web-based QASs, DR is limited to feeding search engines keywords and getting documents with higher ranks. In PR, the paragraph or section containing the likely response is recognized based on the density of keywords.

For the Intelligent QAS proposed by Ahmed et al. [22], to retrieve information, the authors process three submodules which are: Knowledge search module: The QA system is a web-based system. It can handle questions from any field. The system utilizes the Google search engine to find relevant documents. The top 10 documents are then processed. The selected answer is added to the database. Temporal Inference module: Processing questions with temporal inference relies on identifying events and entities, their order in the question and document texts, the temporal characteristics of entities, and recognizing the expected answer type and how it relates to the question's or the candidates' responses' temporal expressions. Additionally, the module for repeated questions: This module implements a semantic similarity-based automatic approach to answering repeated queries. The technique seeks to solve the issue of generating answers for frequently asked questions. The answers and questions are stored in a database. The

current question is matched with previously answered questions using its semantic features.

In LEMAZA [23], Lemur, a retrieval module, was utilized for the PR component. Lemur is an IR toolkit that aids in IR research language, text mining, and modeling. It supports cross-lingual retrieval, has built-in Arabic language support, offers various indexing methods for collections, focuses on PR instead of whole DR, functions with both organized and unstructured DR, and offers C++, Java, and C# APIs.

Otherwise, the IR module in AlQuAnS [24] has two sub-modules: Online Search Engine and PR. The system interfaces with common search engine modules but uses Yahoo API for numerical comparison. The authors use the Explicit Semantic Analysis (ESA) approach proposed in (Gabrilovich and Markovitch, 2007). Furthermore, Lahbari et al. [27] integrate Google as a search engine and Arabic Wikipedia as a dataset.

For SOQAL [28], the process starts with a Document Retriever module, that gathers pertinent documents for the question where they begin by preprocessing each document. The NLTK Arabic tokenizer (Bird, 2006) is used to tokenize and stem the document, and the stop words are removed. The aim of this module is to choose the relevant documents, thus reducing the search for the reader. They use a TF-IDF document retriever for its efficiency.

In AQAS [6], the DR module employs Latent Semantic Indexing (LSI) to extract relevant documents, focusing on paragraphs that contain the answer. The process of retrieval in Latent Semantic Analysis (LSA) makes use of the truncated singular value decomposition (SVD) algorithm. The primary steps involved in constructing the LSA model include preprocessing the documents that have been collected through stemming, splitting of composite words, and removal of stop words. Subsequently, a frequency matrix is created by building the term-document matrix (TDM). Following this, weight functions are employed to improve the efficiency of the IR process and assign weights to terms based on their frequencies. These weight functions include the utilization of Local Weight Function (LWF) and Global Weight Function (GWF). The LWF determines the frequency of a specific word within a text, while the GWF analyzes the frequency of a term across all documents. After this stage, the initial variables undergo decomposition, and queries are projected using LSA for IR

For EWAQ [29], the processing phase incorporates the entailment relation by utilizing AWN with adaptations for the Arabic language. EWAQ evaluates the similarity in entailment between the why-question and passages retrieved through search engines, subsequently re-ranking the retrieved passages based on their entailment similarity.

In the work of Almiman et al. [31], the proposed method extracts and combines various similarity features for ranking question-answer pairs. Lexical similarity features were obtained by establishing a direct correlation among the terms, while semantic similarity features considered context similarity. Char-based similarity features, and PCA-based feature selection were also used. The authors employed a deep neural network model and an ensemble of classification, regression, and BERT models. In [32], the suggested QA system acquires the necessary information for generating responses, utilizing an encoded database containing TEI specific to the domain of hadith narrators. Utilizing a hierarchical structure aid in pinpointing and choosing the necessary components. The operational process relies on the query received to pinpoint the essential elements. Through the stage of Information Extraction, the system can explore the information within the pertinent elements. Subsequently, it refers back to the relevant text to ensure precise delivery of answers. For Biltawi et al. [1], IR techniques utilize various tools such as Lucene, JIRS, and Lemur modules for DR. Additionally, language-independent tools like LingPipe, Protege, and NooJ linguistic engine are also employed.

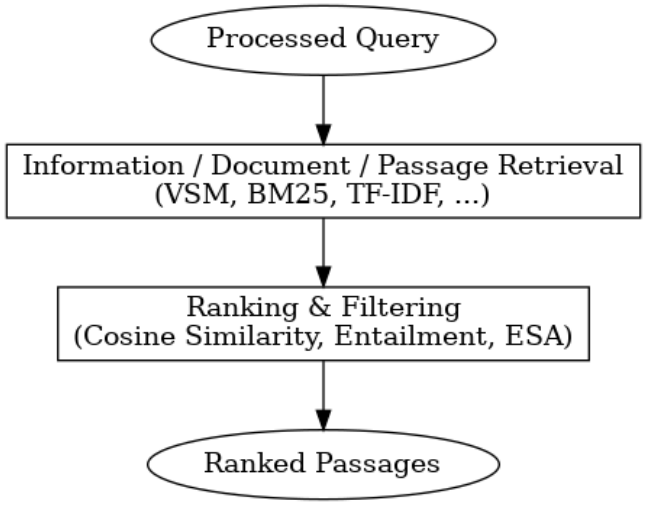

Fig. 9. IR pipeline.

For what concerns IR in [46], Alsubhi et al. use Dense Passage Retriever (DPR) through dense representations for relevance calculation. Neural network encoders are fed text input through dense techniques. A dual encoder with two BERT base models is used in the model architecture. The dot product similarity ranks documents. The passage encoder indexes passages using FAISS. In [36], document research and passage extraction are processed. Most QAS extract many documents for a given question. These systems aim to find a balance between documents and exact answers. They generate passages with specific words or variations of words from the question. This requires an exact response from a large set of documents. The document search module uses Google to find closely related documents based on keywords. Finding pertinent passages that address the questions is the aim. The extractor needs to identify the appropriate passages that hold the solutions. Google is given the question's keywords to find pertinent passages. Search engines like Yahoo and Google employ keywords to locate documents. Google is used in the relevant text passage search method to locate documents or passages that may provide answers to the questions. Text passages serve as crucial connectors between documents and responses. The answer may be revealed by the question marks that appear in the passages. Certain QAS employ specialized search engines such as Lucene or Indri. The authors also use linguistic analysis of questions and text passages as well as post-processing of text passages. First, the text is cleaned and converted from html format to txt format. Then, the text is normalized for analysis. Finally, segmentation divides the

text into sentences and words. Then, in order to analyze questions or text from the Web, linguistic processing involves multiple steps, including NER, syntax analysis, and morphological analysis, which enhance the likelihood of obtaining exact responses in natural language. The authors employ a Logic representation approach to ascertain the relation of textual entailment between questions and passages. They propose using conceptual graphs to find the best-suited logical representation for each question or text passage, based on the Φ operator of Sowa (Sowa John 1984) and they identify the relation of textual entailment between these representations.

In another study, Elkomy and Sarhan [37] use a Post-Processed Ensemble of BERT-based models. Whereas Othman et al. [39] use word embedding learning. Word embeddings can effectively detect similarities between words. They used the Continuous Bag-of-Words (CBOW) model in their word embedding learning module, which outperformed the Skip-gram model on their datasets. They also use LSTM due to its established effectiveness in managing sequential data of varying lengths. A pair of sentences is compared using the Siamese Manhattan LSTM to determine whether they are semantically equivalent. Additionally, they use the Attention Mechanism to calculate each word's probability and assess its overall influence on the question.

TABLE III
IR OR DR OR PR METHODS

Study	Techniques & Tools
[8]	IR system / *Salton's vector space model*; Document processing: tokenization, stop words removal, root extraction, and term weighting.
[9]	Finding relevant passages for the anticipated response.
[10]	Collecting the most relevant snippets from the Web search engine. Snippets containing the question topic are selected using a heuristic.
[11]	IR system / *Numerous database relations*; Document processing: tokenization, stop words removal, root extraction, and term weighting.
[12]	Retrieving and ranking passage / D*istance density N-gram-based PR tool.*; Comparing question and passage representations / *Conceptual Graphs by the new semantic reasoning level.*
[13]	Splitting documents into paragraphs and retrieving the top 5 paragraphs / *an indexing scheme.*
[14]	Generating top-ranked passages / Google search engin*e & JIRS;* Stemming / *Khoja stemmer*.
[15]	Searching for relevant answers / *keyword matching in information sources and Arabic NER.*
[16]	Classifying Quran verses / *A Quranic ontology; M*apping natural language text to Quranic concepts / *A Semantic Interpreter;* Selecting the top scoring verses related to the user question */ Cosine similarity between the input query and every verse in the Quran.*
[17]	Identifying the subset components of the retrieval system, which includes terms from the assumed query; Returning the most probable documents containing the answer; Analyzing the documents*;* Generating closely related answers / Q*uestion classification description.*
[18]	Extracting text from the top 10 retrieved documents / *VSM;* Choosing the top three documents.
[19]	Retrieving relevant information from a knowledge base or document collection based on the user's query.
[21]	Identification and ranking of relevant documents; PR: identifying the paragraph or section containing the possible answer based on the density of keywords
[22]	Knowledge search module */ Google search engine*; Temporal inference module; Repeated question module / *semantic similarity*.
[23]	PR / *Lemur toolkit.*
[24]	Online search engine and PR / *Yahoo API & ESA approach.*
[27]	Search engine / *Google* on Arabic Wikipedia.
[28]	Preprocessing each document: tokenization, stemming / *the NLTK Arabic tokenizer*, and stop words removal; Choosing the relevant documents */ a TF-IDF document retriever.*
[6]	Retrieving relevant documents with selected paragraphs containing the answer / *LSI: SVD algorithm.*
[29]	Implementing the entailment relation / *AWN;* Re-ranking retrieved passages / *entailment similarity*.
[31]	Ranking question-answer pairs */ Lexical similarity features, Semantic similarity features, Char-based similarity features, and PCA-based feature selection; A* deep neural network model and an ensemble of classification, regression, and BERT models are used.
[32]	IR system / *encoded database with TEI for the hadith narrator domain.*
[1]	DR / *JIRS, Lucene, and Lemur modules*; Language-independent tools / *LingPipe, Protege, and NooJ linguistic engine.*
[46]	DPR / *a dual encoder with two BERT base models;* Ranking documents / *the dot product similarity*; Passage encoder / *FAISS.*
[36]	Documents research and passages extraction / *Google; The* extractor; post-processing of text passages; Linguistic analysis of questions and passages: *Text cleaning, transformation from HTML format to txt format, normalization and segmentation;* Linguistic processing: *NER, syntax analysis, and morphological analysis;* Logic representation / *textual entailment & conceptual graphs.*
[37]	PR: a post-processed ensemble of BERT-based models.
[39]	Word embedding learning: detecting similarities between words / *CBOW model, LSTM & Siamese Manhattan LSTM;* Attention mechanism.
[38]	Pretraining BERT / *MLM and NSP;* Tokenization; Generating word embeddings; Fine-tuning and learning query-to-context segmentation and positional embeddings.
[41]	PR: the BM25 model based on the generation-augmented query.
[42]	Word embedding / *ELMo, bi-directional LSTM, QLSTM, semantic similarity.*
[43]	Initializing and Fine-tuning several BERT-like transformers.
[47]	DR / *Using Google API;* PR / *BM25 approach; Retrieving* top-ranked passages / S*entence embedding representation; C*reating vector representations for queries and their associated passages / *pre-trained models: AraBERT, Elmo, and FastText;* Determining similarities between queries and passages / *Soft cosine similarity; E*xtraction top-ranked passages from retrieved passages / *BM25 model.*
[44]	Training two Word2Vec models from Aravec2.0 / *CBOW architecture and SG architecture*; Merging and padding; Extracting textual features / *LSTMs;* Features fusion / *element-wise multiplication*.
[45]	IR: BiLSTM architecture with four layers.

For pretraining in [38], BERT uses two unsupervised learning schemes: Next Sequence Prediction (NSP) and Masked Language Modeling (MLM). In the subsequent step, the training data's question-passage pairs undergo tokenization. Word embeddings are created using the Word Piece tokenizer, which also adds unique tokens to denote the beginning and end of the input sequence. The model learns and refines location embeddings and query-to-context segmentation by using the word embeddings. Furthermore, the retriever module in [41] prepares the source documents from the Arabic Wikipedia dump by removing diacritics and kashidas, normalizing letters, segmenting and tokenizing using the Farasa segmenter, and removing stop words; it also expands queries using pre-trained language models such as mT5-small, AraGPT2-base, and AraGPT2-large. Relevant passages are retrieved by the retriever module using the BM25 model based on the generation-augmented query.

Alkhurayyif and Sait [42] affirmed that ELMo surpasses FastText, Glove, and Word2Vec techniques for word embedding by utilizing bi-directional LSTM to generate sentence vectors. It is capable of producing multi-word embeddings for words in different contexts, assisting in managing the intricacies of Arabic language understanding in NLP applications. The ELMo architecture enables the vectorization of Arabic content. ELMo employs QLSTM to parse queries and search the dataset based on semantic similarity, with query vectors matched in the query module and returned to the response-matching module for further processing. In another work, Alruqi and Alzahrani [43] initialize and Fine-tune many Arabic BERT-like transformers for the extractive QA task that use big datasets of annotated QA pairs. Whereas in [47], two retrieval methods are used for DR and PR. The first method involves using Google API to obtain Arabic Wikipedia articles, and then retrieving passages using the conventional BM25 method. The second method involves using sentence embedding representation for IR. The top-ranked passages obtained from BM25 are retrieved using an Arabic PR module based on sentence embedding. Pre-trained models like AraBERT, Elmo, and FastText are used to constitute vector representations for queries and their associated passages. Soft cosine similarity is employed to determine the similarities between queries and passages. The authors extract the top-ranked passages from the retrieved passages by the BM25 model. Furthermore in [44], two Word2Vec models from Aravec2.0 are used, one with CBOW architecture and one with SG architecture. The models are trained on numerous Arabic topic areas. Merging and padding are performed. Textual features are extracted using LSTMs are used. For feature fusion, element-wise multiplication is conducted. Finally, in [45], BiLSTM architecture with four layers is used.

- **Third component: Answer Extraction**

Through the examination of all QAS proposed in the studies reviewed, in this section, we will answer RQ4: What techniques are applied in the AE component?

The process of answering questions in QARAB [8] consists of four steps. Firstly, the query is processed. Secondly, QARAB converts the query into a "bag of words" and sends it to the IR system. Thirdly, it identifies the expected answer type. Finally, it generates the answer. In QASAL [9], the answer is extracted from the passages while considering the constraint of the Question Analysis module. For every kind of inquiry, the AE module operates distinctly, based on the expected answer by the user. This third module filters the number of tokens in the context using the concordance table. Then, NooJ's facilities are used to export the selected token sequences. The result is saved in a text format or an XML file. For DefArabicQA [10], the AE module is the most crucial module in a definitional QA system, and it is composed of two sub-modules: a candidate definition identification module and a candidate definition filtering module. In the first step, the authors extract candidate definitions using lexical patterns from a collection of snippets. These patterns are manually created and do not involve NLP. A candidate definition is identified if the context surrounding the question topic matches a specific pattern. The identified candidate definitions are then filtered using heuristic rules based on annotated examples. Then, the authors process a "definition ranking" using a statistical approach to rank the candidate definitions based on a global score. Three factors are combined to determine this score: word frequency, snippet position, and pattern weight. The user is shown the top five candidate definitions ranked. The scores from the three criteria are combined as part of the criteria aggregation process.

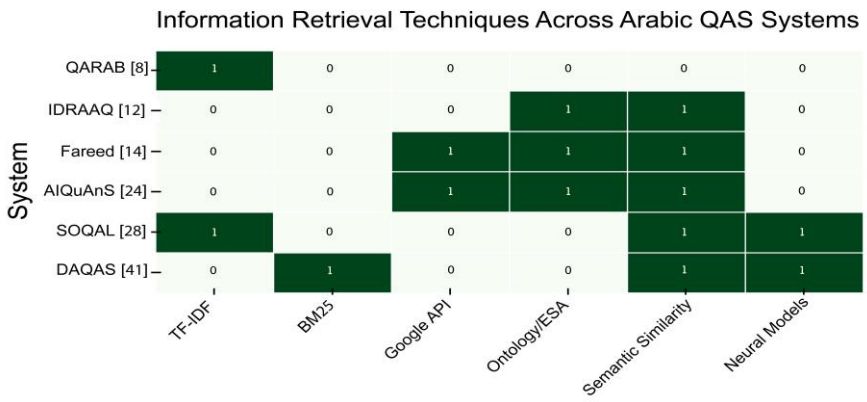

Fig. 10. IR techniques heatmap.

Regarding QArabPro [11], the AE module includes candidate extraction and answer selection. Whereas in IDRAAQ [12], there is an answer validation module that aims to validate answers by using passages provided by the previous module. Otherwise, the answer selection and generation in [13] consists of selecting the best sentence to represent the answer. For the initial phase of the AE module in [14], Fareed et al. extract the answer based on the question type. By searching within highly ranked passages about a person, they can determine the answer to questions about a person's identity. In JAWEB [15], the AE component selects and ranks the most relevant answers. While in Al-Bayan [16], the authors used Arabic NER which involves constructing training data and using it to create a NER model and is then used to tag the input text. Feature extraction involves using various features to calculate the probability of correctness, including the number of matched words, the type of expected answer, Is-A relationships, the count of named entity types, and the minimum distance between matched terms. In [17], the "Answer Selection" module chooses the most accurate answers from a specific type of phrases provided by the Question Analysis (Ferret and al, 2001). This module returns the selected answers to the user, which are obtained from ranked documents based on correctness (Dumais and al, 2002). In [18], the extraction of answers is realized by dividing the raw text into sentences and words. POS tags are then used for named entity detection. Various techniques are applied to extract different types of answers. The question focus is used in the answer selection and ranking stage to choose the top 5 responses.

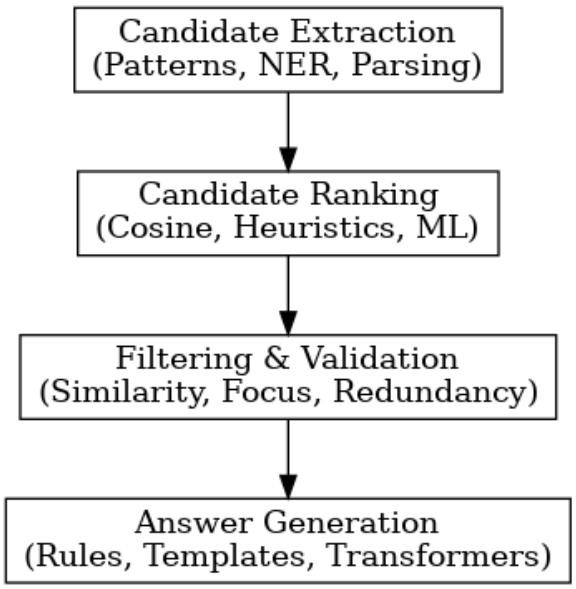

Fig. 11. AE pipeline.

For the AE module in [19], relevant information is extracted using NER and text snippet extraction techniques. Furthermore, a semantic reasoning module is added sometimes. Some QAS employ logical reasoning and inference to derive answers from retrieved information. Ray and Shaalan [21] affirmed that the final part of a QAS is answer processing, which includes identifying candidate answers, ranking them, and formulating the answers. Parsing the extracted passages and comparing them to the anticipated answer type yields candidate answers. These answers are then ranked using algorithms or heuristics. The ranking process assigns weights to candidate answer sentences and filters out those scoring below a pre-established threshold. The remaining sentences are ranked based on their scores. Different strategies can be used to identify and rank answers, such as matching named entities or syntactic relations. In order to match the user's question, the answer formulation process reorganizes the answer sentences. Extracting succinct potential answers is the AE module's primary goal, according to another study [22]. An NER tool initially analyzes the sentences that come from the document analysis module in order to extract phrases that fit the expected answer type. If the expected answer type is a named entity, these phrases form the candidate answers. The answer with the highest similarity score is obtained for answers in the "Definition/Description" class. This is done by utilizing Cosine similarity to calculate the similarity between the question words and the re-ranked passages. The AE module depends on locating the question focus or other cues from the question analysis module. A candidate's score is based on how closely their response to the main theme of the question is similar. The candidate responses with the highest scores are retrieved. Data redundancy gives an answer greater credibility when it appears multiple times in different documents, which is why it is used to verify the accuracy of the response.

The approach in LEMAZA [23] uses Rhetorical Structure Theory (RST) to extract and generate answers to why questions. RST is a leading theory in computational linguistics and has been effectively utilized for NLP applications. The authors use a Rhetorical parsing algorithm to analyze Arabic text and produce a rhetorical structure. They make sure that the keywords from the why-question correlate to one text unit and the answer to another, with these two units being held together by an RST relation. In AlQuAnS [24], Nabil et al. employ a NER system, utilizing extracted patterns for each question type, to adapt the approach proposed by Ravichandran and Hovy (2002). The AE module consists of three stages. The initial stage involves constructing a pattern table for each question type, using web documents retrieved by the PR module. The second stage ranks these patterns based on their precision. The final stage involves finding and filtering answers using the extracted answer patterns and the MADAMIRA NER [25].

In another study, upon constructing the ontology and generating the OWL file, the model in [26] correlates the keywords in the questions with the ontology dictionaries to create SPARQL queries. The authors store all the questions, keywords, and SPARQL queries in an XML file for

utilization in the ontology. The Jena platform executes the SPARQL query to obtain the query answer from the published ontology dataset.

Otherwise, SOQAL [28] contains a machine reading comprehension module that derives answers from the retrieved documents, and an answer ranking module that uses the reader's and document retriever's scores to rank the answers according to relevancy. The proposed reader is BERT, a language model that is currently leading in the SQuAD leaderboard. BERT's core model is a bi-directional Transformer. The input text is tokenized and embedded using a shared vocabulary, with Arabic diacritics removed. Each input instance is represented as a single sentence. For the ranking module, the text documents are given a score by the retriever. The scores for paragraphs are the same as their document. For their hierarchical TF-IDF retriever, the scores are cosine similarities between the document and the question. To obtain potential responses, the document reader is fed the paragraphs obtained from the retriever.

TABLE IV
AE METHODS

Study	Techniques & Tools
[8]	Identifying the expected answer type; Generating the answer.
[9]	Filtering the number of tokens in the context / *concordance table;* Exporting the selected token sequences / *NooJ's facilities;* Saving the result / *text format or an XML file.*
[10]	Extracting candidate definitions / *lexical patterns*; Identifying and filtering candidate definition / *heuristic rules;* Definition ranking / *a statistical approach, a global score combining pattern weight, snippet position, and word frequency;* Presenting the top-5 ranked candidate definitions to the user.
[11]	Candidate extraction and answer selection.
[12]	Validating answers.
[13]	Selecting the best sentence to represent the answer.
[14]	Extracting the answer based on the question type; Searching within highly ranked passages.
[15]	Selecting and ranking the most relevant answers.
[16]	NER model; Tagging input text; Extracting features / *probability of correctness, number of matched words, type of expected answer, Is-A relationships, named entity types, distance between matched terms.*
[17]	Ranking documents / *correctness.*
[18]	Dividing the raw text into sentences and words; NER / *POS tags; E*xtracting different types of answers; Selecting and ranking answers / *question focus to select from the top 5 answers.*
[19]	Extracting relevant information / *NER and text snippet extraction techniques;* Semantic Reasoning module: Deriving answers from retrieved information / *Logical reasoning and inference.*
[21]	Identifying candidate answers / *parsing the retrieved passages and comparing them to the expected answer type*; Ranking them: assigning weights to candidate answer sentences / *algorithms or heuristics; M*atching named entities or syntactic relations*;* Formulating answers: restructuring the answer sentences.
[22]	Analyzing the sentences retrieved from the document analysis / *NER tool; E*xtracting candidate answers / *Cosine similarity; I*dentifying question focus or other clues; Retrieving the highest scoring candidate answers; Validating the accuracy of the answer / *Data redundancy.*
[23]	Extracting and generating answers for why-questions / *RST;* Analyzing Arabic text and producing a rhetorical structure / *Rhetorical parsing algorithm, RST relation.*
[24]	Constructing a pattern table for each question type; Ranking patterns / *precision; F*inding and filtering answers / *MADAMIRA NER.*
[26]	Correlating the keywords in the questions with the ontology dictionaries to create SPARQL queries; Storing questions, keywords, and SPARQL queries / *XML file;* Executing the SPARQL query to obtain the query answer / *Jena platform.*
[28]	Extracting answers from the documents retrieved / *machine reading comprehension module;* BERT as a reader; Ranking the answers / *hierarchical TF-IDF retriever; cosine similarities.*
[30]	Re-ranking the top five passages / *entailment similarity values;* Dividing text into individual sentences; Determining the degree of entailment similarity / *entailment algorithm; I*dentifying the correct answer / *entailment similarity.*
[32]	Predetermining response patterns; Generating a precise and comprehensible reply from the extracted text information.
[1]	AE / *Rule-based approaches, SVM, and Neural Networks.*
[46]	AE: *AraELECTRA passage reader.*
[36]	Locating the specific passage that provides an exact and accurate answer /*a logical textual implication;* Extracting the correct answer /*confidence score.*
[37]	AE: A Post-Processed Ensemble of BERT-based models.
[38]	Generating two vocabulary-sized vectors representing the probabilities of each token as potential start and end positions of the answer span in the context.
[39]	The reader module employs BERT-based models.
[41]	AE: *Several semantic embedding models;* Predicting answers or responses based on surrounding context / *DistilBERT and BERT-based models for Arabic.*
[42]	Fine-tuning AraBERT to get answers from a set of retrieved passages / *next-sentence prediction & masked language modeling.*
[43]	Answer prediction / *MLP of two fully connected layers.*
[45]	Word embedding / *BiLSTM architecture with four layers,* training the model / *backpropagation and the Adam optimizer*.

In EWAQ [30], the AE module operates in the following manner: firstly, the top five passages are re-ranked based on their entailment similarity values. Secondly, if the retrieved passage contains multiple sentences, the text is divided into individual sentences using periods as a basis. Thirdly, the same entailment algorithm mentioned earlier is utilized to determine the degree of entailment similarity between each sentence in each relevant passage and the why question. Finally, the highest degree of entailment similarity is identified as the correct answer. Furthermore, the last stage in the suggested QAS in [32] is answer processing. In this stage, the system generates a precise and comprehensible reply from the extracted text information. The process relies on predetermined response patterns to construct clear sentences with the desired value for the user. The composition procedure employs concatenation, overseen by a verb conjugation checker, to incorporate verbs into the response sentence. In [1], rule-based approaches, SVM, and Neural Networks are techniques used for AE. Whereas in [47], Alsubhi et al. utilize AraELECTRA passage reader for

AE. AraELECTRA is a model for a pre-trained Arabic language representation on large Arabic text corpora utilizing the RTD (Antoun, Baly & Hajj, 2021) approach.

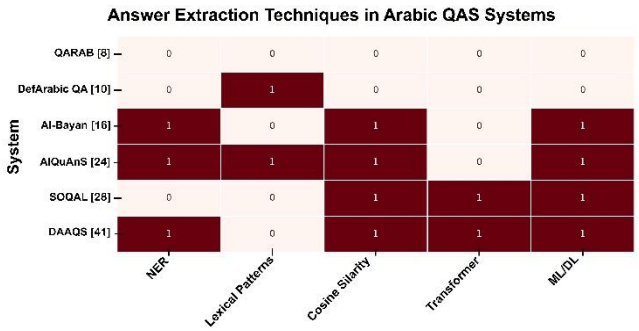

Fig. 12. AE techniques heatmap.

On the other hand, the objective of the search in [36] is to locate the specific passage that provides an exact and accurate answer by examining all the relevant passages, and this process can be represented using a logical textual implication, where passages with a "true" entailment determination are considered as potential answers, and ultimately, the correct answer is determined by selecting the passage with the highest confidence score. Otherwise, Elkomy and Sarhan [37] use a Post-Processed Ensemble of BERT-based models. In [38], the model produces two vocabulary-sized vectors that indicate the likelihood of each token being the beginning or end point of the answer span in the given context. Whereas the reader module in [41] employs BERT-based models. Several semantic embedding methods were applied in [41] to extract answers. Semantic embedding is an NLP technique that represents words or phrases as vectors in a multi-dimensional space, allowing models like BERT and DistilBERT to predict answers based on surrounding context in Arabic.

Lahbari and El Alaou [42] improved AraBERT by using masked language modeling and next-sentence prediction to extract replies from a series of recovered passages. A Multi-Layer Perceptron (MLP) with two fully linked layers is employed in VAQA [43] for Answer prediction. Finally, A. Alazzam et al. [45] process word embedding using BiLSTM architecture with four layers. They trained the model using a three-stage training process based on backpropagation and the Adam optimizer.

IV. Discussion

Based on the reviewed literature, we will present in this section the findings of our SLR following the general flowchart of a QAS: Question Analysis, IR and AE.

We start with the question analysis component. Preprocessing techniques include tokenization, which breaks questions into individual words for easier analysis, as seen in systems like QARAB [8], QASAL [9], and DAQAS [41]. Stop words removal, a standard process in systems such as Yes/No QAS [13], LEMAZA [23], and AlBayan [16], eliminates insignificant words like prepositions and conjunctions.

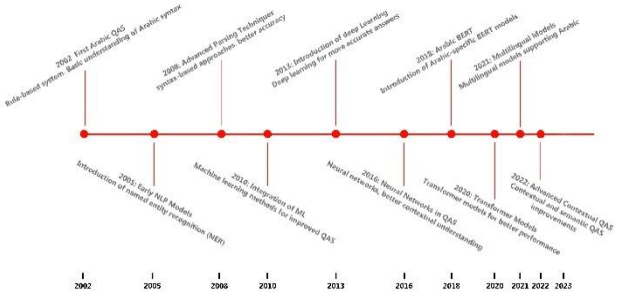

Fig. 13. Timeline of Arabic QAS (2022-2023).

Normalization, used by Al-Bayan [16], SOQAL [28], and others, standardizes question forms by removing diacritics and punctuation. Stemming and lemmatization, essential for handling Arabic morphology, reduce words to their root forms in systems like QARAB [8] and VAQA [44].

POS tagging is widely used in QAS like QARAB [8] and AlQuAnS [24] to assign syntactic categories, helping identify crucial keywords for determining the expected answer type. NER is another key technique employed in systems like QASAL [9] and JAWEB [15], focusing on identifying and categorizing important entities such as names, dates, and organizations. Query expansion, found in QARAB [8] and LEMAZA [23], enriches queries by adding semantically related terms, often using external resources like AWN to improve retrieval.

Question classification is frequently handled using SVM, as seen in Al-Bayan [16] and SOQAL [28], while deep learning methods like LSTM and BERT are used in recent systems like DAQAS [41] and Faris to classify questions and detect duplicates. Focus detection, utilized by systems like DefArabicQA [10] and Yes/No QAS [13], helps identify the core subject of the question, directing the system toward the relevant answer type. Answer type identification, common in QARAB [8] and LEMAZA [23], relies on predefined taxonomies to categorize questions (e.g., factoid, yes/no, list) and align them with the expected answer format

Additionally, ontology-based approaches, used in systems like [26], build domain-specific ontologies to interpret questions based on structured knowledge bases. Machine learning and transformer models, such as BERT in SOQAL [28] and DAQAS [41], significantly improve question analysis performance. Transfer learning techniques, like AraELECTRA [34], also enhance question analysis by leveraging pre-trained models to manage complex tasks.

Regarding the IR module for Arabic QAS, the key techniques include the VSM, which is employed in systems like QARAB [8] and Ahmed and Anto's [18] systems to rank documents based on the query. Techniques such as term weighting, Tokenization, root extraction, and stop words removal are widely utilized across systems like QARAB [8] and QArabPro [11] to enhance document representation.

PR is heavily used in systems like QASAL [9], DefArabicQA [10], and IDRAAQ [12], where relevant passages or snippets are retrieved to reduce the search space. Systems like IDRAAQ [12] incorporate keyword-based and structure-based retrieval, combining semantic processing for

keyword retrieval and N-gram models for structure-based approaches.

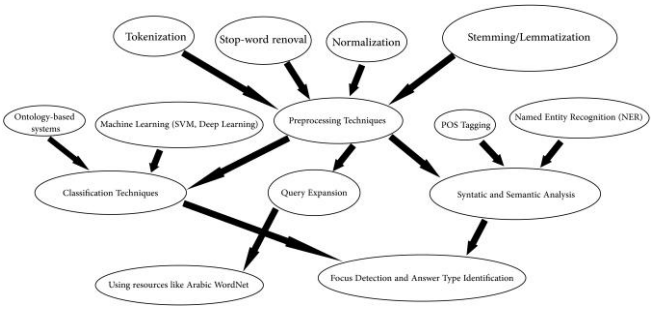

Fig. 14. Flowchart of the main techniques in the Question Analysis component of Arabic QAS.

Some systems, such as [14], [42] integrate external search engines like Google and Yahoo for DR, sometimes enhanced by tools like JIRS for better ranking.

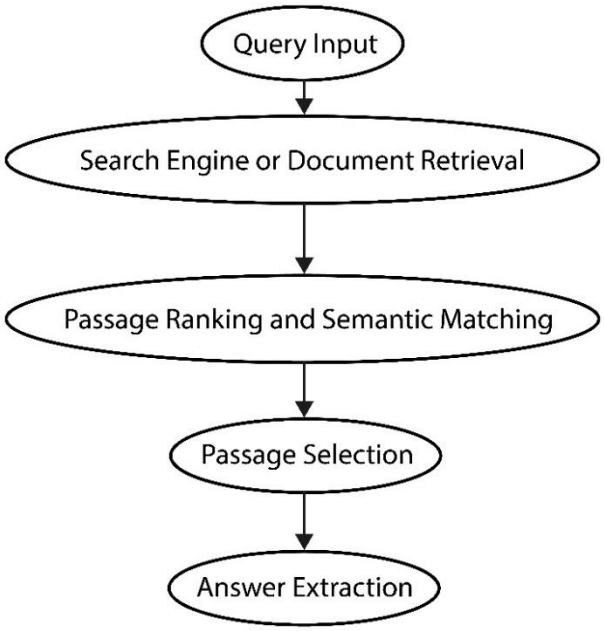

Fig. 15. PR workflow.

LSI, used in AQAS [6], improves DR using techniques like SVD. Advanced systems (e.g., [39] and [43]) utilize neural networks and machine learning models like BERT, BiLSTM, and Siamese Manhattan LSTM for encoding passages and calculating similarity. DPR, found in [46], employs dense neural networks for PR, while word embedding models such as ELMo, BERT, and Word2Vec are used in several systems (e.g., [42], [43], [21]) to better capture context in Arabic text.

Figure 16 illustrates the architecture of a neural network-based IR system, starting with the input of questions and documents into the system. Word embeddings, such as BERT or Word2Vec, convert the text into numerical vectors, followed by dual encoders that separately process the question and passage. Similarity between the encoded passage and question vectors is then calculated using methods like dot product or cosine similarity. Finally, the system ranks the passages based on their similarity to the query, selecting the top-n most relevant passages for further processing.

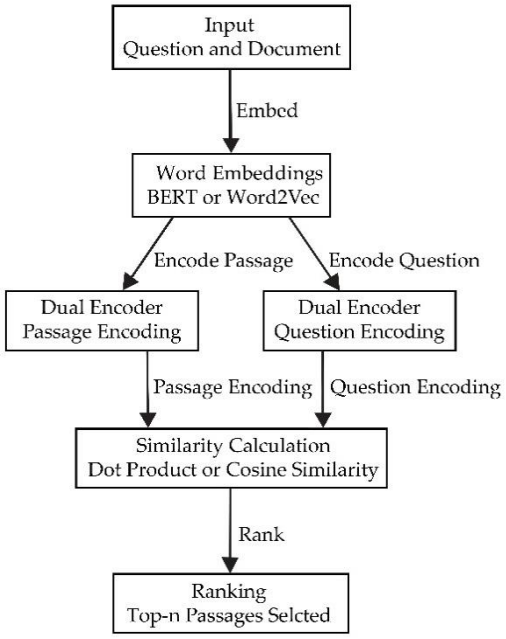

Fig. 16. Neural Network diagram for a BERT-based retrieval model or DPR.

Additionally, ESA used in AlQuAnS [24], computes semantic similarity to enhance DR, while hybrid systems like SOQAL [28] integrate methods such as TF-IDF, stemming, and tokenization to achieve better retrieval results. Various components involved in the IR module include tokenization and stemming, commonly found in systems like SOQAL [28] and DefArabicQA [10], and PR, implemented in systems like IDRAAQ [12] and Yes/No Arabic [13] QAS, where cosine similarity or keyword density determines the relevance of retrieved passages. Several systems, like QArabPro [11], store roots, verbs, and documents in relational databases. Snippets and PR, found in systems like [14], further help reduce the search space, often leveraging external search engines for initial DR.

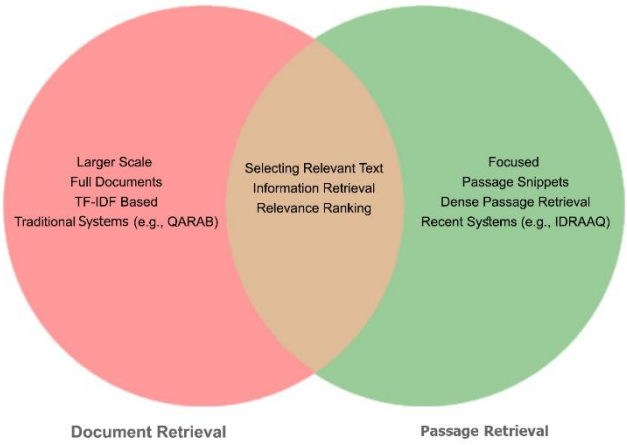

Fig. 17. PR vs DR comparison.

The challenges in IR include the trade-off between PR and DR. Systems like IDRAAQ [12] prioritize PR, where failure to retrieve relevant snippets affects the downstream

AE process. Ranking methods like cosine similarity and TF-IDF play a crucial role in improving document relevance, while semantic matching is another challenge, requiring techniques like conceptual graphs and ELMo embeddings to maintain consistency between the query and the retrieved passages. The complexity of the Arabic language is addressed using tools like the Khoja Stemmer, LSTMs, and modern language models like BERT. Systems like AQAS [6] tackle the issue of data representation using sophisticated weight functions and matrix decomposition to ensure precise DR.

Figure 18 outlines a PR process where the system begins by receiving a question and fetching relevant passages. Both the question and passages are converted into vector embeddings using techniques like BERT or Word2Vec. Similarity between these embeddings is calculated using metrics such as cosine similarity or Dot product. Based on the similarity scores, the system ranks the passages and selects the most relevant ones. Additionally, the system can integrate with external search engines like Google or Wikipedia to further enhance PR.

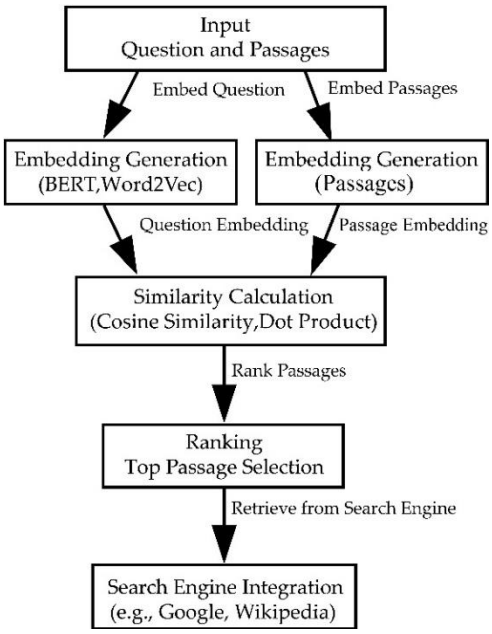

Fig. 18. Embedding-Based IR architecture.

In summary, the IR module is a vital component in Arabic QAS, incorporating a wide range of techniques from traditional methods like VSM and TF-IDF to modern neural approaches like BERT and DPR.

PR plays a key role in narrowing down the search space, while external search engines, word embeddings, and semantic reasoning further enhance the quality of retrieval. Systems address challenges like document ranking precision, Arabic morphology, and balancing passage versus DR through a combination of heuristic, statistical, and neural methods.

In the third component, AE, various techniques are employed by Arabic QAS to extract answers from passages or documents.

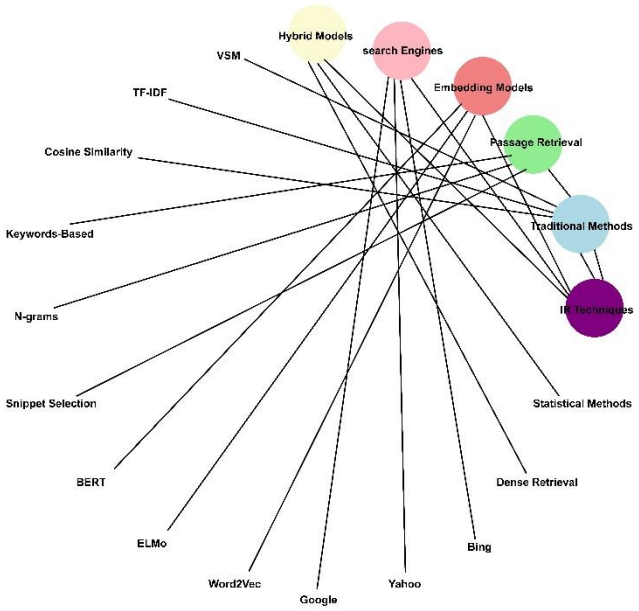

Fig. 19. IR techniques overview diagram.

These techniques span multiple strategies and methods: Candidate extraction and ranking is the first step in the process, where potential answers are identified using patterns found within the text (QARAB [8], DefArabicQA [10], LEMAZA [23]). These candidate answers are then ranked based on several factors, such as relevance to the question, similarity, and the expected answer type (JAWEB [15], SOQAL [28], Al-Bayan [16]). Ranking often involves statistical approaches, including cosine similarity and pattern weighting, which help determine the most appropriate answer (DefArabicQA [10], SOQAL [28]).

Figure 20 illustrates the AE process, starting with the identification of candidate answers using predefined patterns or algorithms. These candidates are then filtered through heuristic methods and pattern matching to remove irrelevant options. After filtering, the remaining candidates are ranked based on relevance scores, which take into account factors like similarity to the question and expected answer types. Finally, the top-ranked answers are selected as the final output.

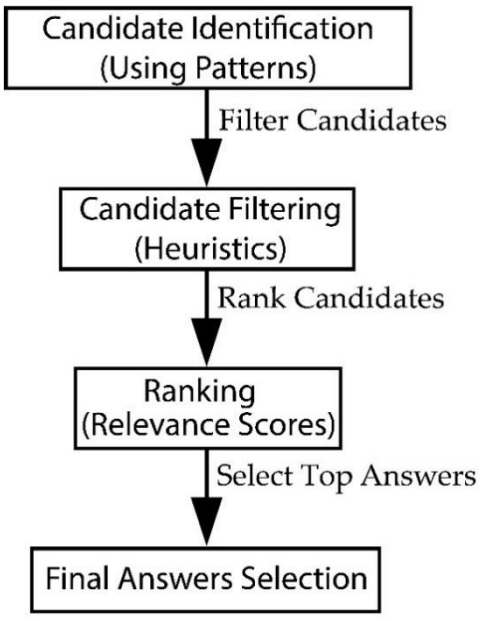

Fig. 20. Candidate AE and ranking workflow.

Furthermore, NER plays a crucial role in many systems, as it identifies and extracts entities that match the expected answer types (Al-Bayan [16], LEMAZA [23], AlQuAnS [24]). In some systems, NER performance is enhanced through additional training data and feature extraction processes to improve the accuracy of entity identification (Al-Bayan [16], SOQAL [28]).

Figure 21 outlines the NER process, beginning with the collection and preparation of labeled training data. Features such as word embeddings or syntactic features are then extracted from this data. Using these features, the core NER model is trained. Once trained, the model outputs the identified entities from the text. These identified entities are then integrated into various QAS architectures for AE and further processing.

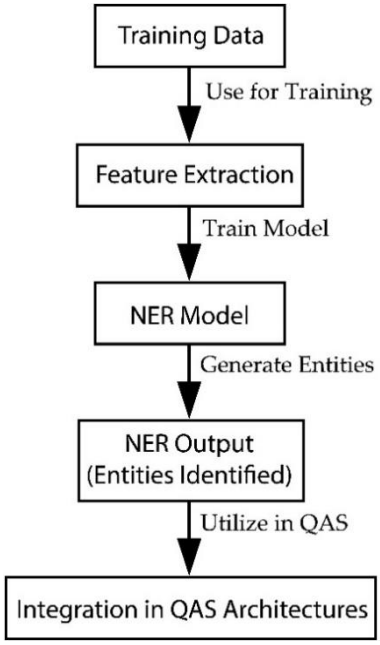

Fig. 21. NER Process.

Moreover, machine learning and NLP models have gained significant traction in modern QAS systems. Techniques like BERT and AraBERT are employed for a deeper semantic understanding and more effective AE (SOQAL [28], EWAQ [30], VAQA [44]). Additionally, models such as BiLSTM, MLP, and various transformer-based architectures are used for more advanced predictions and the generation of complex answers (VAQA [44], [45], LEMAZA [23]).

The diagram illustrated in figure 22 begins with input tokenization, where the text is broken into smaller units like words or sub words. These tokenized inputs are then passed through an embedding layer, converting them into dense vectors that capture semantic meaning. Various deep learning models are employed in the process: BERT, a transformer-based model known for its powerful contextual embeddings; AraBERT, an Arabic variant of BERT tailored for Arabic text processing; BiLSTM, a bidirectional RNN that processes sequences in both directions; and MLP used for classification or regression. Finally, the prediction layer makes predictions based on the processed embeddings and input.

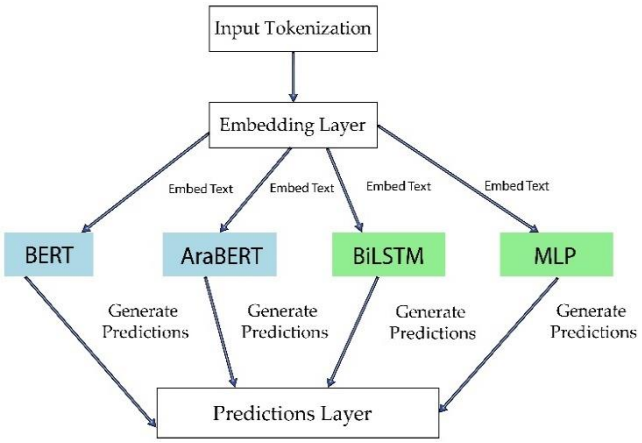

Fig. 22. Deep learning models for AE: Architecture overview.

Besides, logical and semantic reasoning is leveraged in more advanced systems, enabling them to employ logical inference, entailment similarity, and semantic reasoning to deduce answers, especially for more complex questions (EWAQ [30], DefArabicQA [10], LEMAZA [23]). Semantic embedding methods help these systems understand the deeper context of the questions and identify the correct answers more accurately (VAQA [44], AlQuAnS [24]).

The diagram illustrated in figure 23 starts with the input of a question and related passages. The system then evaluates the similarity between the question and passages using entailment techniques. Logical reasoning is applied to interpret the entailment similarity and the context of both the question and passages. Confidence scores are generated based on logical reasoning and similarity analysis to assess the reliability of potential answers. Finally, the system selects the most appropriate answer based on these confidence scores and reasoning.

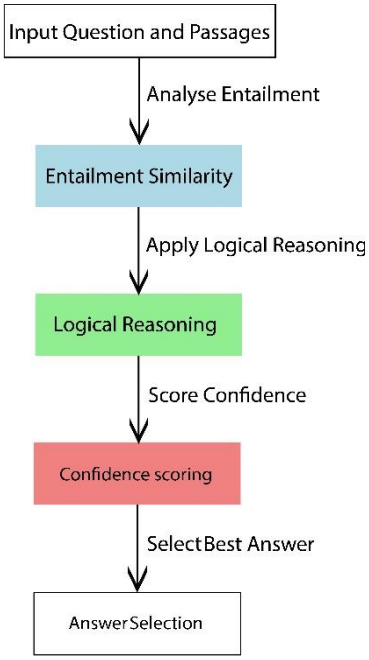

Fig. 23. Logical and semantic reasoning workflow for answer selection.

Finally, answer filtering and validation ensure the quality and accuracy of the extracted answers. Heuristic and rule-based filtering techniques are widely used to eliminate

irrelevant answers and retain only the most accurate ones (DefArabicQA [10], [21], [14]). Additionally, some systems validate answers through redundancy across documents, where multiple appearances of the same answer across various sources reinforce its credibility [21].

The diagram depicted in figure 24 begins with a set of initial candidate answers that need filtering and validation. The first step filters answers based on similarity scores, removing less relevant options. Next, redundancy is checked to ensure answers are not repeated across documents. The remaining answers are then validated for correctness using heuristics or algorithms to ensure accuracy. Finally, the system selects the most accurate answers based on the filtering and validation process.

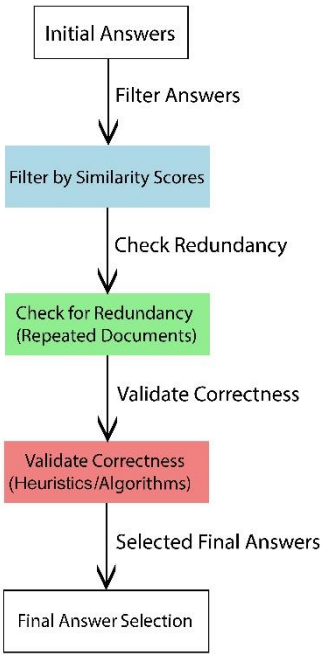

Fig. 24. Answer filtering and validation workflow for answer selection.

In summary, AE in Arabic QAS integrates a range of techniques, from statistical ranking and NER to advanced machine learning models and semantic reasoning, ensuring that the systems extract accurate and contextually relevant answers. After analyzing these results, we were able to categorize Arabic QAS into several classes according to the approach and techniques used in the different components of the system. We present this classification as follows:

TABLE V
ARABIC QAS CATEGORIES

Category	Study
Keyword-based approaches	[8], [11].
Linguistic analysis and NLP techniques	[9], [10], [12], [13], [16], [15], [18], [22], [23], [24], [29].
Semantic and logical-based approaches	[36], [44], [14].
Ontology-based approaches	[14], [26], [6], [42].
Deep learning approaches	[39], [38], [31], [42], [35], [34], [46], [47], [45], [43].
Ensemble models	[37].

Another point is the optimization of an Arabic QAS. For the purpose of improving its accuracy, the following modules or sub-modules can be added to a standard Arabic QAS as shown in Figure 25.

The question classification is a very crucial component of a QAS. In fact, by classifying questions into different categories or types, the system can better understand the user's intent and determine the appropriate strategy for finding the answer. For example, questions may be classified as factoid (requiring a concise factual answer), list (requiring a list of items), or yes/no questions. This classification helps the system select the most suitable approach for answering each type of question, whether it involves retrieving a specific piece of information, generating a list, or providing a simple yes/no response.

Furthermore, many user queries may be underspecified or ambiguous, lacking sufficient context or keywords to accurately retrieve relevant information. Question expansion techniques aim to address this issue by augmenting the original query with additional terms, synonyms, or related concepts. This expanded query provides the system with more context and increases the chances of retrieving relevant documents or passages. Expansion techniques may include synonym mapping, word embedding, or leveraging external knowledge resources such as ontologies or semantic networks.

By incorporating question classification and expansion into the architecture of a QAS, the system's capacity to comprehend customer inquiries, obtain pertinent data, and provide precise answers can be improved by developers, ultimately improving the overall user experience.

Identical question identification or duplicate question detection module holds significant importance within the architecture of an Arabic QAS for various reasons. Firstly, the identification of identical or highly similar questions aids in the avoidance of redundancy. This is crucial as redundant questions can lead to clutter in the system's database and a waste of computational resources. By detecting duplicates, the system can effectively manage and store a more concise collection of unique questions, thereby enhancing the efficiency of storage and retrieval processes. Secondly, the detection of duplicate questions contributes to the optimization of system resources. This optimization becomes particularly crucial in large-scale QAS, where computational resources need to be utilized efficiently for handling vast amounts of data. By identifying and addressing duplicate questions, the system can focus its processing power and resources on answering unique questions. Furthermore, the detection of duplicate questions enhances the user experience by providing a more streamlined and coherent interaction. Through effective identification and management of duplicate questions, the system can offer users diverse and relevant responses, thereby enriching their overall experience. Moreover, the detection of duplicate questions improves the accuracy of the system. Duplicate questions can lead to inaccuracies in the retrieval and ranking of answers. Treating identical questions separately may result in the retrieval of redundant or conflicting information. By identifying duplicates, the system can consolidate responses and present a more accurate and consistent set of answers. In addition, the

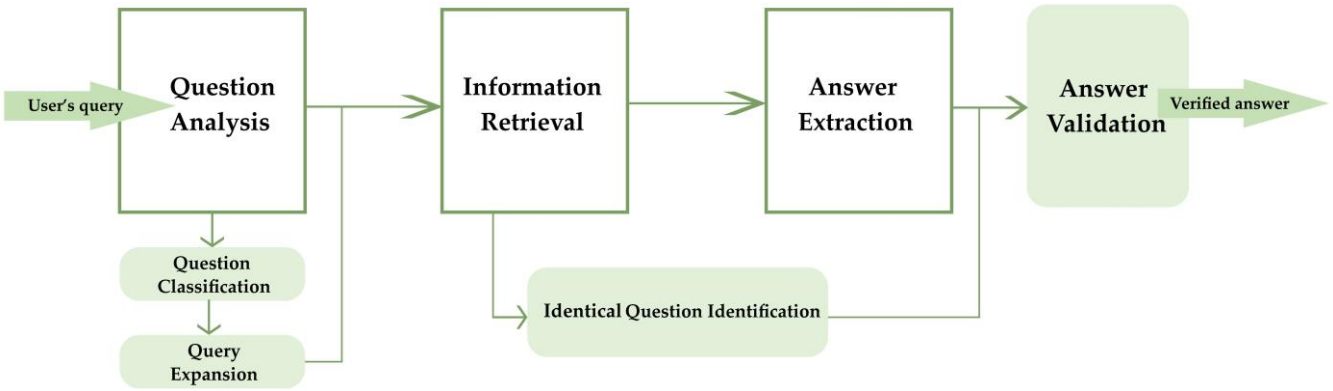

Fig. 25. Improved process of an Arabic QAS.

identification and removal of duplicate questions contribute to the quality of training data in machine learning-based QAS. High-quality training data is essential for the performance and generalization ability of machine learning models utilized in the system. Furthermore, the avoidance of duplicate questions enhances the efficiency of processes such as indexing and searching in systems that rely on a large corpus of documents or passages. Dealing with a reduced set of distinct questions optimizes indexing and searching, leading to faster response times.

In summary, the identification of duplicates in a QAS architecture plays a crucial role in optimizing resources, improving the user experience, maintaining data quality, and enhancing the accuracy and efficiency of the overall system operation.

Finally, the module known as "Answer validation" holds significance in various QAS, including ones designed for the Arabic language, for several reasons. One reason is the assurance of accuracy. The primary objective of a QAS is to provide precise and dependable information. The answer validation module aids in ensuring that the chosen answers are correct and pertinent to the question posed. This is especially crucial in Arabic, where linguistic nuances and context can significantly impact the interpretation of information. Another reason is the filtering of incorrect responses. Not all answers retrieved by a system may be accurate. Some responses may be misleading, outdated, or contextually incorrect. The answer validation module assists in eliminating such incorrect or irrelevant answers, thereby enhancing the overall quality of the system's output. Additionally, the answer validation module helps address ambiguity. Arabic, like any other language, can contain phrases or expressions that are open to multiple interpretations. Answer validation aids the system in cross-verifying potential answers against the context of the question. This is essential for providing users with unambiguous responses. Moreover, the answer validation module plays a role in handling diverse data sources. Arabic QAS often rely on a variety of data sources, each with varying levels of reliability. The answer validation module assists in evaluating the credibility and trustworthiness of information from these sources, enabling the system to prioritize more reliable answers. Furthermore, user trust is crucial for the success of a QAS. Users are more likely to rely on and continue using a system that consistently delivers accurate information. The answer validation module contributes to user trust by ensuring the accuracy and validation of provided answers, thereby enhancing their confidence in the technology. In addition, an answer validation module allows for adaptability to dynamic content. Online content is constantly changing, and information may become outdated over time. By incorporating an answer validation module, the system can adapt to these changes and ensure that the answers provided remain accurate and up to date. Moreover, the answer validation module can be customized for specific domains. In certain applications, such as specific-domain QAS in fields like medicine or law, validating answers based on specific-domain knowledge and rules is crucial. Customization of the answer validation module can be done to meet the requirements of specific domains, thereby improving the system's performance in specialized contexts.

In summary, the answer validation module is a critical component in Arabic QAS. It contributes to accuracy, ensures relevance, and builds user trust by filtering and validating responses based on linguistic, contextual, and domain-specific considerations.

V. Conclusion and Upcoming Works

In this study, we conducted a systematic literature review of the architectures and techniques employed in Arabic QAS, following PRISMA guidelines. We analyzed 40 studies from databases such as Scopus and IEEE Xplore, structuring our analysis around three key components: Question Analysis, IR, and AE.

The question analysis phase demonstrated a range of preprocessing techniques such as tokenization, stop-word removal, and NER, which were essential in improving system understanding and query expansion. Additionally, advanced models like BERT and its Arabic variant AraBERT showed promise in enhancing performance for complex question types. For the IR component, a variety of methods were utilized, from traditional approaches like VSM and TF-IDF to modern deep learning techniques like DPR and embedding models such as Word2Vec and BERT. The integration of search engines and the use of hybrid methods also played a significant role in improving DR and PR accuracy. The AE module presented several approaches, including candidate extraction and ranking, logical reasoning, and validation methods, ensuring the reliability of final answers. Machine learning techniques, particularly deep learning models such as BERT and AraBERT, have been effective in understanding and extracting precise answers from Arabic texts. Our findings demonstrate that a variety of approaches—such as ensemble models, deep learning, ontology-based, semantic/logical reasoning, and keyword-based methods—are used in developing Arabic

QAS. Crucial sub-modules like question classification, question expansion, identical question identification, and answer validation were identified as pivotal in enhancing system accuracy.

Overall, Arabic QAS has shown considerable advancement over the years, particularly with the introduction of deep learning models. However, challenges remain, especially in handling the complexity of the Arabic language, such as morphology and syntax. Future research will focus on further integrating these components to improve system performance and robustness. We aim to enhance semantic understanding, leverage larger datasets, and incorporate more robust reasoning mechanisms. Additionally, we plan to evaluate several techniques across different datasets and use the results to develop a comprehensive Arabic QAS for specific domains.

REFERENCES

[1] M. M. Biltawi, S. Tedmori, and A. Awajan, "Arabic question answering systems: Gap analysis", IEEE Access, vol. 9, pp. 63876–63904, 2021, doi: 10.1109/ACCESS.2021.3074950.

[2] I. Srba and M. Bielikova, "A comprehensive survey and classification of approaches for community question answering", ACM Trans. Web, vol. 10, no. 3, p. 18:1-18:63, Aug. 2016, doi: 10.1145/2934687.

[3] I. V. Serban et al., "Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus", arXiv, May 29, 2016. doi: 10.48550/arXiv.1603.06807.

[4] A. Moubaiddin, O. Shalbak, B. Hammo, and N. Obeid, "Arabic dialogue system for hotel reservation based on natural language processing techniques", Comput. Sist., vol. 19, no. 1, Mar. 2015, doi: 10.13053/cys-19-1-1962.

[5] B. Shawar, "A chatbot as a natural web interface to Arabic web QA", Int. J. Emerg. Technol. Learn., vol. 6, Mar. 2011, doi: 10.3991/ijet.v6i1.1502.

[6] M. A. Al-Shannaq, K. M. O. Nahar, and K. M. K. H. Halawani, "AQAS: Arabic question answering system based on SVM, SVD, and LSI", Journal of Theoretical and Applied Information Technology, vol. 97, no. 2, pp. 681–691, Sep. 2020. [Online]. Available: https://www.jatit.org/volumes/Vol97No2/27Vol97No2.pdf

[7] D. Moher et al., "Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement", Syst. Rev., vol. 4, no. 1, p. 1, Jan. 2015, doi: 10.1186/2046-4053-4-1.

[8] B. Hammo, H. Abu-Salem, S. Lytinen, and M. Evens, "QARAB: A question answering system to support the Arabic language", in Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages, Association for Computational Linguistics, Philadelphia, USA, Jul. 2002, pp. 1–11, doi: 10.3115/1118637.1118644.

[9] W. Brini, M. Ellouze, S. Mesfar, and L. H. Belguith, "An Arabic question-answering system for factoid questions", in 2009 International Conference on Natural Language Processing and Knowledge Engineering, Sep. 2009, pp. 1–7. doi: 10.1109/NLPKE.2009.5313730.

[10] O. Trigui, L. H. Belguith, and P. Rosso (2010). "DefArabicQA: Arabic Definition Question Answering System", in Proceedings of the 7th Workshop on Language Resources and Human Language Technologies for Semitic Languages (pp. 40–45). Valletta, Malta. [Online]. Available: https://personales.upv.es/prosso/resources/TriguiEtAl_LREC10.pdf

[11] M. Akour, S. Abufardeh, K. Magel, and Q. Al-Radaideh, "QArabPro: A rule-based question answering system for reading comprehension tests in Arabic", American Journal of Applied Sciences, vol. 8, no. 6, pp. 652–661, Jun. 2011, doi: 10.3844/ajassp.2011.652.661.

[12] L. Abouenour, K. Bouzoubaa, and P. Rosso, "IDRAAQ: new Arabic question answering system based on query expansion and passage retrieval", in CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, 2012, pp. 1–12. [Online]. Available: https://ceur-ws.org/Vol-1178/CLEF2012wn-QA4MRE-AbouenourEt2012.pdf

[13] W. Bdour and N. Gharaibeh, "Development of yes/no Arabic question answering system", Int. J. Artif. Intell. Appl., vol. 4, Feb. 2013, doi: 10.5121/ijaia.2013.4105.

[14] N. S. Fareed, H. M. Mousa, and A. B. Elsisi, "Enhanced semantic Arabic question answering system based on Khoja stemmer and AWN", in 2013 9th International Computer Engineering Conference (ICENCO), Dec. 2013, pp. 85–91. doi: 10.1109/ICENCO.2013.6736481.

[15] H. Kurdi, S. Alkhaider, and N. Alfaifi, "Development and evaluation of a web-based question answering system for Arabic language", in Proceedings of the International Conference on Computer Science & Information Technology (CSIT)*, Feb. 2014, pp. 187–202. doi: 10.5121/csit.2014.4216.

[16] H. Abdelnasser et al., "Al-Bayan: An Arabic question answering system for the Holy Quran", Jan. 2014. doi: 10.13140/2.1.4113.4400.

[17] H. M. A. Chalabi, "Question Processing for Arabic Question Answering System", M.S. thesis, Faculty of Engineering, The British University in Dubai, Dubai, UAE, 2013. [Online]. Available: http://bspace.buid.ac.ae/handle/1234/760

[18] W. Ahmed and B. Anto, "Question analysis for Arabic question answering systems", Int. J. Nat. Lang. Comput. IJNLC, vol. 05, pp. 21–30, Dec. 2016, doi: 10.5121/ijnlc.2016.5603.

[19] W. Bakari, P. Bellot, and M. Neji, "Researches and reviews in Arabic question answering: Principal approaches and systems with classification", in Proceedings of the International Arab Conference on Information Technology (ACIT)*, 2016. Accessed: Mar. 18, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Researches-and-Reviews-in-Arabic-Question-Answering-Bakari-Bellot/ba54db29088759e0d3b33a974726cc289e81f6fe

[20] M. Shakir and M. Sheker, "Domain-specific ontology-based approach for Arabic question answering", Technical Report, ResearchGate, Feb. 2016. [Online]. Available: https://doi.org/10.13140/RG.2.1.3276.1369

[21] S. K. Ray and K. Shaalan, "A Review and Future Perspectives of Arabic Question Answering Systems", IEEE Trans. Knowl. Data Eng., vol. 28, no. 12, pp. 3169–3190, Dec. 2016, doi: 10.1109/TKDE.2016.2607201.

[22] W. Ahmed, A. Dasan, B. Anto, and A. Pv, "Developing an intelligent question answering system", Int. J. Educ. Manag. Eng., vol. 6, pp. 50–61, Nov. 2017, doi: 10.5815/ijeme.2017.06.06.

[23] A. Azmi and N. Al-Shenaifi, "Lemaza: An Arabic why-question answering system", Nat. Lang. Eng., vol. 23, pp. 1–27, Aug. 2017, doi: 10.1017/S1351324917000304.

[24] M. Nabil, A. Abdelmegied, Y. Ayman, A. Fathy, G. Khairy, M. Yousri, N. El-Makky, and K. Nagi, "AlQuAnS – An Arabic language question answering system", in Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KDIR 2017), vol. 1, 2017, pp. 144–154. doi: 10.5220/0006602901440154.

[25] A. Pasha et al., "MADAMIRA: A Fast, comprehensive tool for morphological analysis and disambiguation of Arabic", in Proc. 9th Int. Conf. Lang. Resour. Eval., pp. 1094–1101, May 2014.

[26] A. Albarghothi, F. Khater, and K. Shaalan, "Arabic question answering using ontology", Procedia Comput. Sci., vol. 117, pp. 183–191, Jan. 2017, doi: 10.1016/j.procs.2017.10.108.

[27] I. Lahbari, S. E. Alaoui, and K. Zidani, "Toward a new Arabic question answering system", International Journal of Speech Technology, vol. 15, no. 3, pp. 1–9, 2018, doi: 10.1007/s10772-018-9501-9.

[28] H. Mozannar, K. Hajal, E. Maamary, and H. Hajj, "Neural Arabic Question Answering", in Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, Aug. 2019, pp. 108–118, doi: 10.18653/v1/W19-4612.

[29] S. Romeo et al., "Language processing and learning models for community question answering in Arabic", Inf. Process. Manag., vol. 56, no. 2, pp. 274–290, Mar. 2019, doi: 10.1016/j.ipm.2017.07.003.

[30] F. T. AL-Khawaldeh, "Answer extraction for why Arabic questions answering systems: EWAQ", arXiv, Jul. 04, 2019. doi: 10.48550/arXiv.1907.04149.

[31] A. Almiman, N. Osman, and M. Torki, "Deep neural network approach for Arabic community question answering", Alex. Eng. J., vol. 59, no. 6, pp. 4427–4434, Dec. 2020, doi: 10.1016/j.aej.2020.07.048.

[32] H. Maraoui, K. Haddar, and L. Romary, "Arabic factoid question-answering system for Islamic sciences using normalized corpora", Procedia Comput. Sci., vol. 192, pp. 69–79, Jan. 2021, doi: 10.1016/j.procs.2021.08.008.

[33] A. Hamza, N. En-Nahnahi, K. A. Zidani, and S. El Alaoui Ouatik, "An Arabic question classification method based on new taxonomy and continuous distributed representation of words", J. King Saud Univ. - Comput. Inf. Sci., vol. 33, no. 2, pp. 218–224, Feb. 2021, doi: 10.1016/j.jksuci.2019.01.001.

[34] D. Premasiri, T. Ranasinghe, W. Zaghouani, and R. Mitkov, "DTW at Qur'an QA 2022: Utilising Transfer Learning with Transformers for Question Answering in a Low-resource Domain", arXiv, May 12,

2022. Accessed: Mar. 17, 2024. [Online]. Available: http://arxiv.org/abs/2205.06025

[35] H. Faris, M. Habib, M. Faris, A. Alomari, P. Castillo, and M. Alomari, “Classification of Arabic healthcare questions based on word embeddings learned from massive consultations: A deep learning approach”, J. Ambient Intell. Humaniz. Comput., vol. 13, Apr. 2022, doi: 10.1007/s12652-021-02948-w.

[36] W. Bakari and M. Neji, “A novel semantic and logical-based approach integrating RTE technique in the Arabic question–answering”, Int. J. Speech Technol., vol. 25, no. 1, pp. 1–17, Mar. 2022, doi: 10.1007/s10772-020-09684-0.

[37] M. Elkomy and A. M. Sarhan, “TCE at Qur’an QA 2022: Arabic Language Question Answering Over Holy Qur’an Using a Post-Processed Ensemble of BERT-based Models”, in Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection (OSACT 2022), Marseille, France, June 2022, pp. 154–161. [Online]. Available: https://aclanthology.org/2022.osact-1.19/

[38] E. Aftab and M. K. Malik, “eRock at Qur’an QA 2022: Contemporary Deep Neural Networks for Qur’an based Reading Comprehension Question Answers”, in Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur’an QA and Fine-Grained Hate Speech Detection, H. Al-Khalifa, T. Elsayed, H. Mubarak, A. Al-Thubaity, W. Magdy, and K. Darwish, Eds., Marseille, France: European Language Resources Association, Jun. 2022, pp. 96–103. Accessed: Mar. 18, 2024. [Online]. Available: https://aclanthology.org/2022.osact-1.11

[39] N. Othman, R. Faiz, and K. Smaïli, “Learning English and Arabic question similarity with Siamese neural networks in community question answering services”, Data Knowl. Eng., vol. 138, p. 101962, Mar. 2022, doi: 10.1016/j.datak.2021.101962.

[40] M. Abdelhafez, G. Khateeb, and A. Yahya, “Efficient Arabic query auto-completion for question answering at a university”, in 2022 International Arab Conference on Information Technology (ACIT), Nov. 2022, pp. 1–9. doi: 10.1109/ACIT57182.2022.9994190.

[41] H. Alami, A. El Mahdaouy, A. Benlahbib, N. En-Nahnahi, I. Berrada, and S. E. A. Ouatik, “DAQAS: Deep Arabic Question Answering System based on duplicate question detection and machine reading comprehension”, J. King Saud Univ. - Comput. Inf. Sci., vol. 35, no. 8, p. 101709, Sep. 2023, doi: 10.1016/j.jksuci.2023.101709.

[42] Y. Alkhurayyif and A. R. W. Sait, “Developing an Open Domain Arabic Question Answering System Using a Deep Learning Technique”, IEEE Access, vol. 11, pp. 69131–69143, 2023, doi: 10.1109/ACCESS.2023.3292190.

[43] T. Alruqi and S. Alzahrani, “Evaluation of an Arabic chatbot based on extractive question-answering transfer learning and language transformers”, AI, vol. 4, pp. 667–691, Aug. 2023, doi: 10.3390/ai4030035.

[44] S. Kamel, S. Hassan, and L. Elrefaei, “VAQA: Visual Arabic question answering”, Arab. J. Sci. Eng., vol. 48, Mar. 2023, doi: 10.1007/s13369-023-07687-y.

[45] B. A. Alazzam, M. Alkhatib, and K. Shaalan, “Arabic educational neural network chatbot”, Inf. Sci. Lett., vol. 12, p. 54, Jun. 2023, doi: 10.18576/isl/120654.

[46] K. Alsubhi, A. Jamal, and A. Alhothali, “Deep learning-based approach for Arabic open domain question answering”, PeerJ Comput. Sci., vol. 8, p. e952, May 2022, doi: 10.7717/peerj-cs.952.

[47] I. Lahbari and S. O. El Alaou, “Exploring Sentence Embedding Representation for Arabic Question/Answering”, Int. J. Comput. Digit. Syst., vol. 15, no. 1, pp. 1229–1241, Mar. 2024, doi: 10.12785/ijcds/150187.