Arab J Sci Eng (2014) 39:4541–4564
DOI 10.1007/s13369-014-1062-2

REVIEW ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Arabic Question Answering: Systems, Resources, Tools, and Future Trends

Mohamed Shaheen · Ahmed Magdy Ezzeldin

Received: 24 November 2012 / Accepted: 15 April 2013 / Published online: 12 April 2014

Abstract Arabic is the 6th most wide-spread natural language in the world with more than 350 million native speakers. Arabic question answering systems are gaining great importance due to the increasing amounts of Arabic content on the Internet and the increasing demand for information that regular information retrieval techniques cannot satisfy. In spite of the importance of Arabic question answering, there is no review that covers Arabic question answering systems, tools, resources, and test-sets so far, which was the motivation for this work. In this survey, different Arabic question answering systems are demonstrated and analyzed and the main question answering tasks like question analysis, passage retrieval, and answer extraction are explored. The main difficulties of modern standard Arabic and how these difficulties are tamed and classified are also explained. Arabic question answering evaluation metrics, test-sets, and language resources are reviewed, and future trends are also highlighted to guide new research in this area. This survey provides guidance for new research in Arabic question answering to get up-to-date knowledge about the state-of-the-art approaches in this area. It also demonstrates the tools created and used by researchers to build an Arabic question answering system.

Keywords QA · Factoid questions · QA4MRE · Question analysis · Passage retrieval · Answer extraction · Answer validation · Test-sets · Evaluation · Metrics · Language resources · NLP · Information retrieval · Stemming · Corpus · NER · Stemming · Lemmatization · Morphological analysis · Part-of-speech tagging · Diacritization · Overview · Review · Survey

الخلاصة

ان اللغة العربية هي اللغة السادسة في العالم من حيث سعة الانتشار، حيث أنها اللغة الأصلية لأكثر من 350 مليون نسمة ، وتزداد أهمية إجابة الأسئلة العربية بزيادة المحتوى العربي على الشبكة العنكبوتية وازدياد الحاجة إلى المعلومات وعدم قدرة أنظمة استرجاع المعلومات التقليدية على إرضاء تلك الحاجة. وبالرغم من أهمية إجابة الأسئلة العربية إلا أنه لا يوجد مسح شامل لأنظمة إجابة الأسئلة العربية وأدواتها ومواردها واختباراتها إلى الآن وهذا هو الدافع وراء هذا العمل. في هذا المسح الشامل سيتم تناول أنظمة إجابة الأسئلة العربية بالتحليل والشرح واستكشاف المهام الرئيسية في نظم إجابة الأسئلة مثل تحليل الأسئلة واسترجاع الفقرات واستخراج الإجابات ، وشرح وتصنيف صعوبات اللغة العربية الفصحى الحديثة وكيفية حل تلك الصعوبات.، وسيتم أيضاَ إلقاء الضوء على الاختبارات ومقاييس التقييم والموارد اللغوية والاتجاهات المستقبلية لإجابة الأسئلة العربية لتوجيه البحث المستقبلي في ذلك المجال. ان هذا المسح الشامل يوجه الأبحاث المستقبلية في مجال إجابة الأسئلة العربية لكي يحصلوا على المعرفة اللازمة عن أفضل الأساليب في هذا المجال ، وهو أيضاَ يقوم بشرح الأدوات المستخدمة من قبل الباحثين لبناء أنظمة إجابة الأسئلة العربية.

1 Introduction

In information retrieval (IR) and natural language processing (NLP), question answering (QA) is the task of automatically providing an answer for a question posed by a human in natural language. QA as a task can be divided into three main distinct subtasks: question analysis, passage retrieval, and answer extraction. Most question answering systems follow these three subtasks; however, they may differ in how they implement every subtask.

Question answering as a problem deals with many types of questions. Factoid questions are one type that is concerned with questions that ask mainly about named entities (NEs) like questions using the words: when, where, how

M. Shaheen · A. M. Ezzeldin (✉)
College of Computing and Information Technology,
Arab Academy for Science, Technology and Maritime Transport,
Alexandria, Egypt
e-mail: a.magdy@a1works.com

M. Shaheen
e-mail: cshaheen@hotmail.com

much/many, who, and what, which ask about a date/time, a place, a person, and an organization, respectively. Question answering for machine reading evaluation (QA4MRE) is another type of QA that evaluates how the computer understands a comprehension passage by posing a list of multiple choice questions that can be answered by understanding this comprehension passage. Another type is the definition questions that ask about the meaning of a term or a concept. Questions that use the words why or how are another type that is hard to answer and there are very little, if any attempts done to answer this type of questions.

QA systems are more capable of handling natural language queries than regular IR systems. On the other hand, regular IR systems like search engines yield better results when the query is in Boolean formula [43]. QA systems are also easier to use and have higher recall than ordinary IR systems, which means that QA systems return an answer if it exists, unlike regular IR systems that sometimes return irrelevant documents which may not contain the answer [60].

In the field of QA, English and other Latin-based languages benefited a lot from the advancement in this field. However, Arabic question answering systems are lagging behind when compared to their English and Latin-based counterparts due to the Arabic-specific difficulties. One of these Arabic-specific difficulties is that traditional Arabic orthography has diacritics, which adds vowel sounds to the Arabic words; however, most modern Arabic documents use an undiacritized version of Arabic that is called modern standard Arabic (MSA). The lack of diacritics in MSA adds a lot of ambiguity to Arabic morphology and semantics. Arabic is also a highly inflectional and derivational language and it has no capital letters to define named entities (NEs). All these difficulties are explained in Sect. 2. The tools and language resources (LRs) used to tame these difficulties are explained in Sect. 3. These reviewed tools include named entity recognition (NER), passage retrieval (PR), logic and inference tools, and morphological analysis toolkits for text normalization, tokenization, part-of-speech (PoS) tagging, diacritization, base phrase chunking (BPC), stemming, and lemmatization.

Section 4 reviews the used Arabic QA evaluation metrics to provide a deeper understanding of the evaluation process of the different Arabic QA systems. In Sect. 5, the different Arabic QA test-sets are reviewed to highlight the transparent, objective ways of evaluating future Arabic QA systems.

Some attempts were made to reach an acceptable result in the Arabic question answering task. Most of these attempts suffered from over-fitting and subjective, non-realistic evaluation. Among the earliest attempts to tackle the Arabic question answering problem was AQAS, an Arabic QA System developed by Mohamed et al. [48]. AQAS used a knowledge-based model that can only search for answers in structured data [48]. From 1993 till 2002 many advancements in the field of Arabic NLP and IR were done that led to the creation of QARAB which was used with unstructured documents written in Arabic for Al-Raya newspaper in Qatar [33,34]. Nevertheless, QARAB evaluation was biased as it only used 113 factoid questions as a test-set and QARAB creators themselves were the evaluating users for the system that they created. It is also skeptical that their evaluation results were much higher than the state-of-the-art work done in English question answering. See Table 4.

In Sect. 6, the most prominent Arabic QA systems and the main Arabic QA tasks are reviewed: question analysis, passage retrieval (PR), and answer extraction. In Sect. 7, the future trends of Arabic QA are highlighted, according to the current trends of Questions Answering systems for other languages that have not been tackled by Arabic QA researchers yet.

2 Arabic-Specific Difficulties

Arabic is a very rich language; however, this richness needs special handling, which makes regular NLP systems, designed for other languages, unable to handle it. Arabic is a highly derivational language as the vocabulary of Arabic words are essentially built from about 10,000 three- or four-letter roots, and derivations of these roots are created by adding affixes (prefix, infix, or suffix) to each root according to about 120 patterns. Derivations in Arabic are almost always like this: lemma = root + pattern [1]. See Fig. 1.

This derivational nature increases the size of the Arabic vocabulary dramatically and makes building a high-coverage semantic language resources (LR) very challenging. However, the highly derivational nature of Arabic has been tamed to a great extent using Arabic morphological analysis tools and LRs discussed in the next section.

Arabic language morphology is challenging when compared to English and other Latin-based languages. This is because Arabic is a highly inflectional language where a word token can consist of multiple morphemes. An Arabic word may take this form “word = lemma + affixes (prefix, infix, and suffix)”. The prefixes can be articles, prepositions, or conjunctions, which causes a lot of sparseness in index documents and makes query expansion harder. See Fig. 2. This

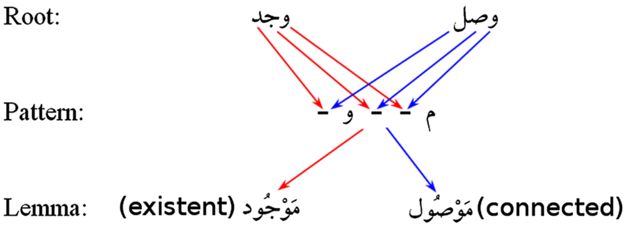


Fig. 1 Example Arabic derivation [1]

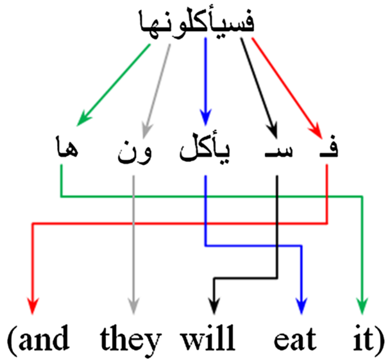


Fig. 2 Example Arabic inflection [1]

inflectional nature needs special handling for different Arabic NLP tasks like stemming, lemmatization, morphological analysis, PoS tagging, and even tokenization. Various tools were developed to address this need as described shortly.

Unlike English and most Latin-based languages, Arabic does not have capital letters which makes named entity recognition (NER) harder. In the next section, we will review the different approaches to the NER task.

One of the Arabic-specific difficulties is the lack of diacritics in modern standard Arabic (MSA), which adds to the ambiguity of the question and the searched documents. For example, the word "علم" in MSA can mean "عَلَم" (Flag) or "عِلْم" (Science) according to context. However, much interest has been given to diacritizing MSA to resolve this ambiguity. The state-of-the-art in this area is the work accomplished by Rashwan et al. [55] that could solve the Arabic diacritization problem with a very small error rate of about 3.1–12.5 %.

Like any other language, Arabic NLP needs language resources (LRs). These LRs like lexicons, corpora, treebanks, and ontologies are essential for syntactic and semantic tasks either to be used with machine learning or for lookup and validation of processed words. In the next section, we will review the Arabic LRs that are important for Arabic QA and its subtasks.

3 Arabic QA Tools

3.1 Morphological Analysis

Morphological analysis tools solve the problems that emerge from the inflectional and derivational nature of the Arabic language and the lack of diacritics in the modern standard Arabic. They are concerned with typical syntactic NLP tasks like

- Tokenization: separation of word morphemes into separate tokens.
- Diacritization: adding diacritics (Tashkeel) to MSA, which disambiguates the meaning.
- Stemming: removing affixes from words.
- Part-of-speech (PoS) tagging: determining the word part of speech (noun, verb, preposition, etc.).
- Lemmatization: returning a word to its root (may depend on PoS tagging).

3.1.1 AraMorph (Java port of Buckwalter Arabic Morphological Analyzer)

AraMorph[1] is another morphological analyzer that is a Java port of the Buckwalter Arabic morphological analyzer (BAMA) which was written in PERL. It has a dictionary-based Arabic Stemmer, and it applies transliteration to the Arabic word based on Buckwalter's transliteration system. So, كتاب is transliterated into ktAb prior to morphological analysis [25]. Obviously, transliteration adds an unneeded performance penalty to the stemming process making it slower. AraMorph then uses a brute-force algorithm to decompose the word in a sequence of possible prefix, stem, and suffix, which makes the stemming process slower. It also marks the semantic features of gender and number when they are indicated by a gender and/or number suffix. It could tag only 13 % of the nouns in a 3000-word corpus and 35.5 % of a 20-million-word corpus [28].

3.1.2 MADA+TOKAN

Habash et al. [32] created MADA+TOKAN a freely available toolkit that offers various Arabic NLP services like tokenization, diacritization, morphological disambiguation, part-of-speech (PoS) tagging, stemming, and lemmatization. MADA examines all possible analyses for each word and then selects the analysis that matches the current context using support vector machine (SVM) model classification for 19 distinct, weighted morphological features. TOKAN takes the output of MADA and generates tokenized output in a customizable format. MADA has over 86 % accuracy in predicting full diacritization [32].

3.1.3 AMIRA Tools

Mona Diab [27] introduced the AMIRA toolkit, which includes a clitic tokenizer, PoS tagger, and base phrase chunker (shallow syntactic parser). The technology of AMIRA is based on supervised learning with no dependence on explicit modeling or knowledge of deep morphology. It also gives the user the flexibility to request tokenized or non tokenized PoS tagged output. The PoS tagger accuracy is 96 %. The

[1] AraMorph: http://www.nongnu.org/aramorph/.

AMIRA tokenizer applies a layer of learning to classify the ending of words after tokenization, so as to set the final t as either t "ت" or taa marbuuta "ة", , and the final A "ا" as either Y alef maqsuura "ى" or not, and it has an *F* measure of 99.2 %.

The AMIRA base phrase chunking (BPC) is the process of grouping adjacent words together to form syntactic phrases such as noun phrases (NPs), verb phrases (VPs), and prepositional phrases (PPs).

e.g. [I]NP [would eat]VP [red luscious apples]NP [on Sundays]PP

BPC is the first step in shallow syntactic parsing, which is very important to semantic analysis in QA. It is also a lot faster than deep syntactic parsing. The AMIRA BPC uses the Arabic Treebank for training and scores an *F* measure of 96.33 % [27].

3.1.4 Fassieh®

Fassieh® is an Arabic text annotation tool by Attia et al. [12]. It can carry out different Arabic language factorizations at high coverage that exceeds 99.8 %. Among these Arabic language factorizations are morphological analysis, PoS tagging, diacritization, and lexical semantic analysis. Fassieh® also resolves the high ambiguity of these language analyses statistically with error rate less than 5 %. It also allows supervised proofreading of these factorizations for error intolerant tasks [12].

3.1.5 Abouenour et al. Morphological Analyzer

Abouenour et al. [3] manually developed a lexicon with explicit linguistic classes and three dictionaries: one for Arabic nouns, one for prefixes, and one for suffixes and integrated them into a new morphological analyzer, based on a new classification of the Arabic nouns and provided useful information for syntax and semantics. It provides morpho-syntactic features for different Arabic morphemes such as number, gender, person, and grammatical functions. They started by tokenizing the Arabic text and finding all possible solutions for each token and then looking up each solution in the lexicon. They have also fixed problems like broken plurals and proper names. Their morphological analyzer has an average performance of 82.14 % on a 1.66 GHz core 2 duo processor with 1 GB RAM and 1 MB cache [3]. This analyzer was the integrated into SAFAR. See Sect. 1.

3.1.6 Other Stemmers

As mentioned in the previous section, Arabic is a highly inflectional and derivational language. So, some words may rarely occur with the same form, which causes sparseness in the indexed documents and affects the passage retrieval process dramatically. Some attempts were made to come over the inflectional nature of Arabic by using stemmers. Stemming is the process of removing affixes (prefixes, suffixes and infixes) from words to reduce them to their stems. For example, stemming the English word "computing" produces the root "comput", which is the same root produced by the word "computation" [39].

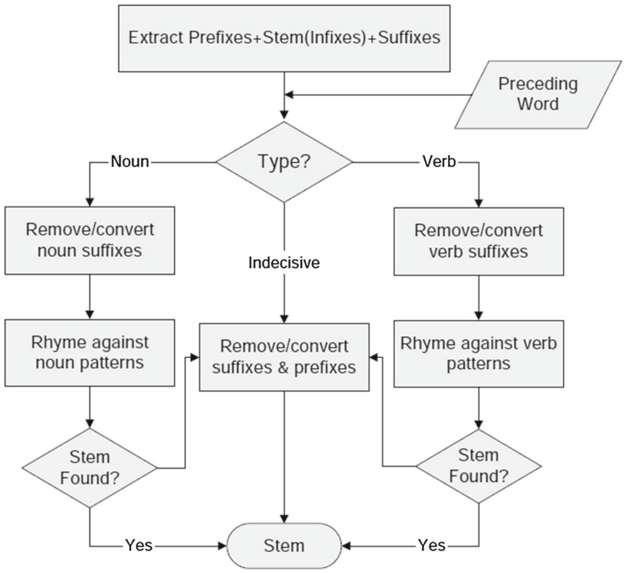

Extract Prefixes+Stem(Infixes)+Suffixes
Preceding Word
Type?
Noun
Verb
Indecisive
Remove/convert noun suffixes
Remove/convert verb suffixes
Rhyme against noun patterns
Remove/convert suffixes & prefixes
Rhyme against verb patterns
Stem Found?
Stem Found?
Yes
Stem
Yes

Fig. 3 Harmanani et al. Arabic stemming approach [35]

AlShalabi [10] removed suffixes and prefixes from inflected word and then matched the resulting word with some patterns without using a dictionary to check the resulting stem by creating rules to define if letters belong to the root or not. He then tested his approach on 10,582 Arabic words and achieved an accuracy of 92 % [10].

Harmanani et al. [35] used a new approach based on language-dependent rules, interpreted by a rule engine. They determine the type of the word if it is a noun or a verb then check if this word rhymes with one of the patterns of this type to extract the stem. They showed the effect of their stemmer on indexing and stated that the indexing precision reached 100 % after indexing 19 documents and that their stemmer enhanced the speed of indexing by 75 % [35]. See Fig. 3.

Kadri and Nie [37] used a statistical approach first to define the most frequent prefixes and suffixes based on the occurrence frequencies of these affixes on the 523,359 different tokens of the TREC collection and they found out that some affixes are the most frequent affixes: see Table 1. They obtained a mean average precision (MAP) of 31 % with their stemming method on a merged topics collection against 28 % for the light stemming technique.

Shereen Khoja[2] [39] implemented the Khoja Arabic stemmer in C++ and Java. This stemmer works by removing the

[2] Shereen Khoja research: http://zeus.cs.pacificu.edu/shereen/research.htm.

Table 1 Arabic affixes [37]

Antefixes	Prefixes	Suffixes	Postfixes
وبال ، وال ، بال ، فال ، كال ، ولل ، ال ، وب ، ول ، لل ، فس ، فب ، فل ، وس ، ك ، ف ، و ، ب ، ل	ا ، ن ، ي ، ت	تما ، يون ، تين ، تان ، ات ، ان ، ون ، ين ، وا ، تا ، تم ، تن ، نا ، ت ، ن ، ا ، ي ، و	كما ، هما ، كن ، هن ، تي ، ها ، نا ، هم ، كم ، ك ، ه ، ي
Prepositions meaning respectively: and with the, and the, with the, then the, as the, and to (for) the, the, and with, and to (for), then will, then with, then to (for), and will, as, then, and, with, to (for)	Letters meaning the conjugation person of verbs in the present tense	Terminating of conjugation for verbs and dual/plural/female marks for nouns	Pronouns meaning respectively: your, their, your, their, my, her, our, their, your, your, his, my

longest suffix and the longest prefix, then matches the remaining word with the verbal and noun patterns, to extract the root. Khoja Arabic stemmer handles weak letters (i.e. alif, waw or yah) that may change in the Arabic word root. It also handles Arabic words that do not have roots like the Arabic equivalents of we, after, under, and so on. If a letter is deleted from the root during derivation due to duplicate letters (i.e. the last two letters are the same), the stemmer also handles this issue and produces the right root [39]. Taghva et al. [61] reported that it has an Average Precision of 46.3 %. Khoja Stemmer was also used in an Arabic information retrieval system by Larkey and Connell [41], and they reported that the average precision of their system improved by 49 % over the non-stemmed technique.

Taghva et al. [61] created the Information Science Research Institute’s (ISRI) stemmer, which has many features in common with the Khoja stemmer; however, it does not use a root dictionary. This feature makes ISRI stemmer more capable of stemming rare and new words. It returns a normalized form for unstemmed words and has more stemming patterns and more than 60 stop words. ISRI stemmer has an average precision of 48 % [61].

3.2 Named Entity Recognition

Named entity recognition (NER) in Arabic is harder than in English and other Latin-based languages, due to the lack of capital letters as mentioned in Sect. 2. This forces researches to tackle the NER problem differently in Arabic. NER is also crucial for QA systems as Factoid questions ask about named entities (person, organization, location, date, etc.). In this section we will review the different approaches to Arabic NER.

Abuleil and Evens [8] described a system for building an Arabic lexicon automatically by tagging Arabic newspaper text that depends on the keywords to find proper nouns. Their system was composed of four subsystems [8]:

- Morphology analyzer: to analyze suffixes and prefixes.
- Type-finder system: uses the morphology analyzer and goes through some tests to find the part of speech of the word.
- Feature-finder system: uses the morphology analyzer to find word features (gender, number, person, and tense).
- Database: a lexicon started by a hand-built set of data and had tables for verbs, nouns, particles, and proper nouns.

Benajiba et al. [18] used NER based on combining PoS tagging information with a Maximum Entropy model. They used proper names context information as features and used the ANERsys 1.0 which depends on ANERcorp and ANERgazet that they have created. ANERsys 1.0 has a Precision of 63.21 %, a Recall of 49.04 %, and an *F* measure of 55.23 % [18].

Benajiba and Rosso [15] created ANERsys 2.0 that adopted a 2-step approach to NER to solve the problems of multi-token NEs: (1) step 1 is concerned mainly by detecting the start and the final tokens of each NE, and (2) Step 2 takes care of classifying them. The performance of this version scored a Precision of 70.24 %, a Recall of 62.08 %, and an *F* measure of 65.91 % [15].

Benajiba and Rosso [20] made some experiments on ANERsys that used the maximum entropy model; however, they repeated their experiments using conditional random fields (CRF) with the same features to compare it with the maximum entropy model. Results showed that using both CRF and maximum entropy model together in ANERsys got a precision of 89.20 %, a recall of 54.63 %, and an *F* measure of 67.76 % [20].

Zaghouani et al. [66] used language-independent rules, but they made reference to language-specific words which they called trigger words. The trigger word lists include titles (السيدة Mrs., استاذ Prof., دكتور Dr., etc.), professions or positions (مدير Director, رئيس President, محام lawyer, etc.), country adjectives (التونسي Tunisian, الكندي Canadian, etc.), religious and ethnic groups (الكاثوليكي Catholic, السنية Sunni, بربر Berber). They also introduced a list of modifiers, which are words that can appear in certain places between the name mention and the trigger words. They also added a list of known names to make the task easier. Their NER system performs at a precision of 87.17 %, a recall of 65.74 % and an *F* measure of 74.95 % [66].

Abdelrahman et al. [2] integrated two machine learning techniques which are bootstrapping semi-supervised pattern recognition and conditional random fields (CRF) classifier as a supervised technique. They used pattern and word semantic fields as CRF features. They also applied a sixfold cross-validation and found out that their work outperformed previous CRF work. They used 15 features, which are [2]

- The word itself
- The word part of speech
- Base phrase chunks (BPC) the phrase of the current word whether it is a noun phrase, a verb phrase a prepositional phrase, a conjunctive phrase, an adjectival phrase or an adverbial phrase.
- The existence of the word in the gazetteers
- The first two and last three characters of the word (e.g., "عبد" / “Abd” is a very repetitive prefix in Arabic person names)
- Semantic fields (group similar words in one semantic group)
- The pattern of the word

Morphological features which are

- Suffix and prefix: most Arabic NEs have no suffix or prefix
- Diptote ("الممنوع من الصرف”) many proper names and names of places are diptotes like مصر /Misr and إبراهيم / Ibraheem
- Definiteness “ال” / “the”
- Interjection article, such as “يا” used to call for some one
- Relative pronoun: “الذين ،التي ،الذي ”” / “who, whom, which”
- Nasikh Particle, such as “كان , ان” always needs subject after it and usually the subject is a named entity.
- Interrogation article, such as Hamza "ء"
- Relative adjective “المجموعة الاستثمارية” / “The Investment Group”

They reached an average *F* measure of all NE classes of 73.63 % using a conditional random fields classifier and it was increased to reach 80.60 % by using the bootstrapping semi-supervised pattern recognition technique [2].

3.3 Language Resources (LRs)

Although Arabic language corpora, lexicons, and machine-readable dictionaries were scarce a decade ago, a lot of effort has been made throughout the past decade to solve this problem.

Maamouri et al. [44] introduced the Penn Arabic Treebank (PATB), which is a corpus of manually Arabic parsed sentences to be used by many researchers for training data-driven parsing algorithms [44]. It is very important in training supervised machine learning to accomplish many morphological tasks like tokenization, PoS tagging, shallow and deep syntactic parsing.

Elkateb et al. [29] introduced the Arabic WordNet and described the challenges they faced to create it. Arabic WordNet is a lexical resource for modern standard Arabic based on the widely used Princeton WordNet for English. Arabic WordNet was also enriched by Abouenour et al. [4–6] by adding new named entities, new verbs, and new nouns which enriched the hyponymy relation between concepts.

Benajiba et al. [18] introduced ANERcorp for training and testing and ANERgazet as a gazetteer for named entities (NEs) which had the following gazetteers:

- Location Gazetteer: which consists of 1,950 names of continents, countries, cities.
- Person Gazetteer: a list of 1,920 complete person names
- Organizations Gazetteer: which consists of a list of 262 names of companies, football teams and other organizations. ANERcorp a corpus of Arabic named entities for training and testing [18].

In an attempt to provide a solution for the high derivational nature of Arabic, Attia et al. [11] introduced an Arabic lexical semantics language resources (LR) that enables the retrieval of the possible senses of any Arabic word at a high coverage. This semantic LR relates PoS-tags and morphologically constrained Arabic lexical compounds to a predefined limited set of semantic fields across which the semantic relations are defined.

Mesfar et al. [46] created a lexicon named “El-DicAr” (Electronic Dictionary for Arabic), which is an electronic dictionary that links the morphological and syntactic-semantic information to its list of lemmas. El-DicAr has a coverage of 92.53 %.

3.4 Passage Retrieval Tools

3.4.1 Apache Lucene

Lucene is a keyword-based, open source, Full-Text search library written in Java. It was created by Doug Cutting in 1999 and joined the Apache Software Foundation in 2001. At the time of this writing, Apache Lucene latest version number is 4.1 and it forms the core of many industrial and academic IR systems worldwide [36]. Lucene features Arabic tokenization and normalization filters and an Arabic light stemmer. Lucene implements light stemming as specified by Larkey et al. [42]. It works by removing stop words, definite articles, and (و) “and” from the beginning of words, and a small number of suffixes from the ends of words. It improved the average precision of IR queries by 35.3 % which is higher than the performance of Buckwalter’s Arabic Morphological Analyzer in the same conditions.

3.4.2 JIRS

JAVA information retrieval system (JIRS[3]) is an information retrieval system with special interest in question answering created by Gomez et al. [31]. Unlike the traditional search engines that are based on question keywords, JIRS retrieves passages that will most likely contain the answer. This is achieved by carrying out a search based on question n-grams using three different language independent n-gram based models. It is easy to customize and adapt thanks to its powerful kernel and modular design. It provides a standard passage retrieval system based on the space vectorial model (SVM) and three models based on n-grams: the simple N-gram model (SNM), the term-weight N-gram model (TNM) and the distance N-gram model (DNM) [31].

DNM considers a sequence of n adjacent words (n-gram) extracted from a sentence or a question and assigns a relevance score according to the similarity between the question and the retrieved passages n-grams. JIRS searches all n-grams of the question and assigns them a score according to the n-grams and weight that appear in the retrieved passages. The more this structure is similar to the one of the question, the more relevant the passage is considered [31].

Table 2 Systems that used JIRS as a PR system

QA system	How JIRS was used	Performance
Benajiba et al. [16, 17, 19]	Adapted JIRS for Arabic by creating a light stemmer, Arabic stop words list and normalization rules, and adding support for Arabic text encoding	Coverage (measure of passages containing the right answer): 69 %
Abouenour et al. [4]	Used JIRS beside Yahoo search engine API Used distance density N-gram model to rank passages according to the appearance of question terms nearer to each other Used the stop words list available with JIRS in the question expansion phase	Accuracy increased from 9.66 to 20.20 %

Benajiba et al. [26] adapted JIRS to Arabic question answering by applying light-stemming which caused the coverage measure to be raised up to 69 % and the redundancy measure up to 3.28 [19]. JIRS was proved to be better than Apache Lucene in QA applications thanks to its Distance N-gram Model that ranks retrieved passages more efficiently [26]. Table 2 compares the Arabic QA systems that use JIRS.

3.5 Other Tools

3.5.1 SAFAR (Software Architecture for Arabic Language pRocessing)

SAFAR[4] is an open source modular platform, written in Java. It provides an integrated development environment (IDE) to ANLP (Arabic natural language processing) [58].

As illustrated in Fig. 4, SAFAR layers are developed as a set of reusable Java components [58]:

- Tools: statistical functions and test tools
- Resources Services: Arabic lexicons and corpora
- Basic services: 3 layers of Arabic NLP services (morphology, syntax and semantics)
- Applications: the high-level applications that use the previous layers.
- Client applications: which use the different layers through APIs.

[3] JIRS page on SourceForge: http://sourceforge.net/projects/jirs.

[4] SAFAR: http://sibawayh.emi.ac.ma/safar/.

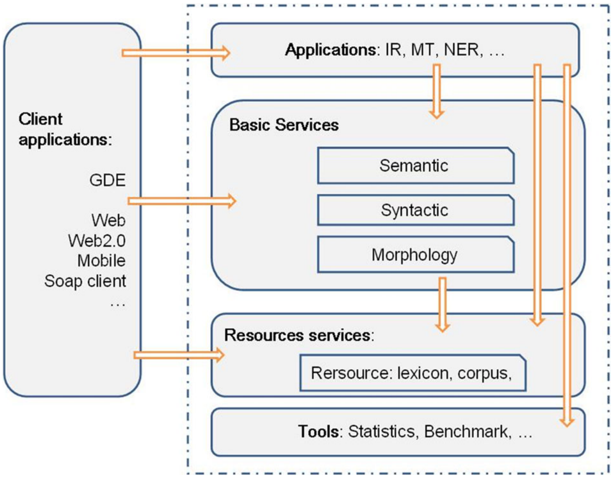


Fig. 4 Architecture of SAFAR [58]

SAFAR morphological analyzer provides services like light stemming, introducing information about stems and their prefixes and suffixes, and finding all possible results existing in the Arabic lexicon. It has an average performance of 82.14 % [3].

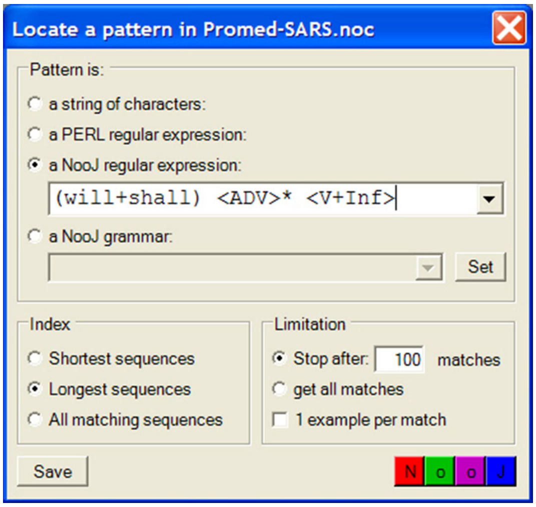
3.5.2 NooJ

NooJ[5] is a freeware linguistic engineering development environment written by Professor Max Silberztein 2002. NooJ is written using C# .NET on the Visual Studio .NET Platform. It can import 100 text file formats. It can also export and import annotations as XML tags. It can use PERL-type regular expressions, NooJ regular expressions and NooJ grammars so that any morphological, lexical, syntactic or semantic information annotated in the text can be used inside NooJ expressions and grammars (see Fig. 5). Examples: “any Human noun in the plural”, “any transitive verb in the infinitive”, “any plural noun phrase”), etc. [59].

NooJ Context-free grammars are recursive transition networks, which allow users to recognize certain sequences of texts and to associate them with annotations. See Fig. 6. NooJ also has tools to test, debug, and maintain these grammars [59].

All NooJ morphological, lexical, syntactic and semantic analyses produce XML annotations. It is also possible to import lexical, syntactic and semantic annotations from, or exported to, XML documents, while NooJ Disambiguation grammars are used to filter out annotations.

NooJ features some Arabic language resources provided by Mesfar Slim from Université de Franche-Comté. These resources are a sample text, a dictionary of 10,000+ verbs, their inflection in the form of a NooJ inflectional grammar, built by Ibtihal Farawi and Mesfar Slim, and a group of morphological grammars for verb prefixes and suffixes.


Locate a pattern in Promed-SARS.noc

Pattern is:
○ a string of characters:
○ a PERL regular expression:
● a NooJ regular expression:
`(will+shall) <ADV>* <V+Inf>`
○ a NooJ grammar:

Index
○ Shortest sequences
● Longest sequences
○ All matching sequences

Limitation
● Stop after: 100 matches
○ get all matches
□ 1 example per match

Fig. 5 A simple expression to identify the future tense

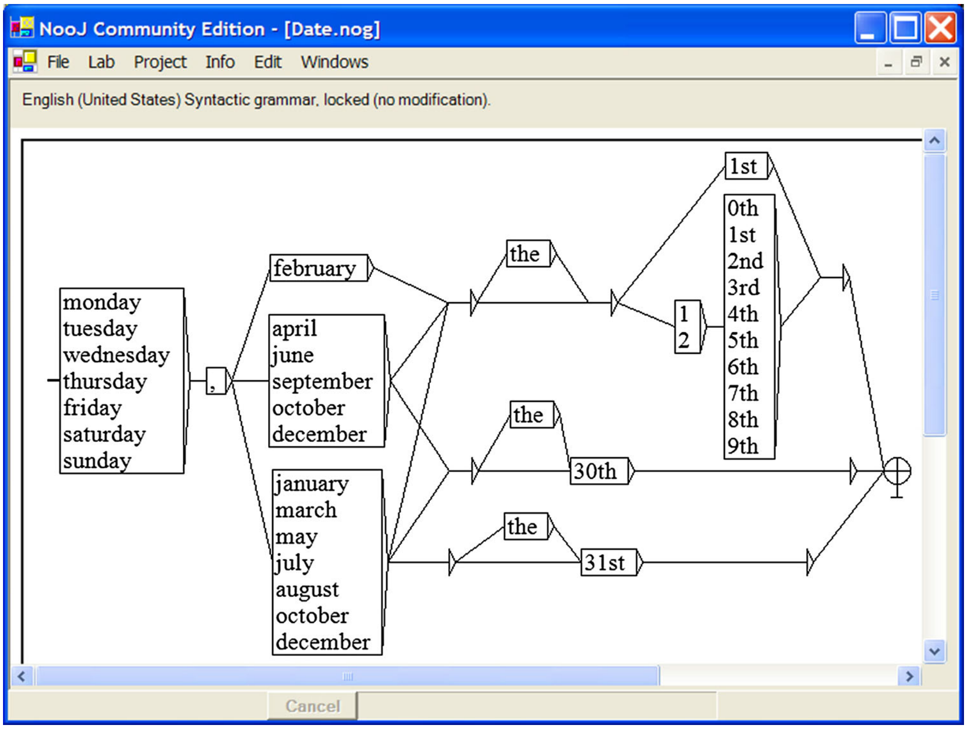
3.5.3 Amine Platform

Amine Platform[6] is a Java platform for intelligent systems and multi-agents which is the result of 20 years of research done by Pr. Adil Kabbaj since 1985 that summed up the work done in the field of conceptual graphs (CG) Theory and AI. Amine platform is of great importance to artificial intelligence and cognitive science, and semantic web. It is used for semantic analysis of questions and answers in QA systems. Amine platform was reported to be usable in the Arabic NLP field [22]. However, there is no reported performance for Amine Platform. It has seven main layers:

- Ontology layer: supports building and manipulating different kinds of ontologies that are based on Concept Structures.
- Knowledge base layer: supports rule-based and case-based knowledge bases. A knowledge base in Amine is a concept graph (CG) with an ontology to support it.
- Algebraic layer: contains different data types and data structures like (AmineInteger, AmineDouble, List, Set, Term, Concept, Relation and CG). It also contains matching-based operations, such as match, equality, unification, subsumption, maximalJoin and generalization.

[5] NooJ official website: http://www.nooj4nlp.net.

[6] Amine platform page on SourceForge: http://amine-platform.sourceforge.net.



English (United States) Syntactic grammar, locked (no modification).

Fig. 6 A simple NooJ graph that identifies dates

- Memory layer: supports inference strategies like induction, deduction, abduction, and analogy
- Programming layer: supports memory-based, rule-based and pattern-matching and activation and propagation-based programming paradigms.
- Multi-agent systems layer: used “Agent Development Environments” for agent creation and manipulation and used Amine for higher cognitive and reactive capabilities.
- Applications layer: creates agents using the other layers.

3.5.4 Stanford NLP

StanfordNLP Toolkit[7] is a group of libraries that cover the most common tasks of NLP and can be used in question answering tasks. The following are some among the libraries included in Stanford NLP Toolkit:

- Stanford Parser: which is an implementation of a probabilistic natural language parser, a dependency parser, and a lexicalized PCFG parser in Java.
- Stanford PoS Tagger: which is a maximum-entropy part-of-speech (PoS) tagger for English, Arabic, Chinese, French, and German, written in Java.
- Stanford Named Entity Recognizer: which is a conditional random field (CRF) sequence model, with a list of features for named entity recognition in English and German.
- Stanford Word Segmenter: which is a CRF-based word segmenter in Java which also supports Arabic and Chinese.
- Stanford Classifier: which is a machine learning classifier for text categorization, a maximum entropy, and multi-class logistic regression model.
- Tregex and Tsurgeon: a utility for matching patterns in trees, and a tree-transformation utility.
- Phrasal: a phrase-based machine translation system.
- Stanford Biomedical Event Parser (SBEP)
- Stanford English Tokenizer
- Stanford Tokens Regex: regular expressions over tokens.
- Stanford Temporal Tagger (SUTime): rule-based temporal tagger for English text.

O’Steen and Breeden [51] used Stanford PoS Tagger together with Buckwalter Arabic morphological analyzer to generate features for their Arabic named entity recognition system. They also reported that the Arabic Stanford PoS tagger performed at 96.72 % accuracy on the development set and 77.49 % accuracy on unknown words.

3.5.5 Open NLP

Apache OpenNLP[8] is a machine learning based library for the processing of natural language text that supports

[7] StanfordNLP group official website: http://nlp.stanford.edu/software/index.shtml.

[8] OpenNLP official website: http://opennlp.apache.org.

many NLP tasks like tokenization, sentence segmentation, PoS tagging, NER, chunking, parsing, maximum entropy, perceptron-based machine learning, and co-reference resolution. Apache OpenNLP consists of many modules: sentence detector, tokenizer, name entity recognizer, document categorizer, part-of-speech tagger, chunker, parser, co-reference resolution module, corpora, and machine learning (maximum entropy) module.

3.5.6 GATE

General architecture for text engineering[9] (GATE) is a Java suite of tools developed at the University of Sheffield started in 1995. Languages currently handled in GATE include English, Spanish, Chinese, Arabic, Bulgarian, French, German, Hindi, Italian, Cebuano, Romanian, and Russian. It can import documents in many formats, such as TXT, HTML, XML, DOC, PDF documents, and Java Serial, PostgreSQL, Lucene, Oracle Databases with help of RDBMS storage over JDBC. GATE as a platform includes an information extraction system called a nearly new information extraction system (ANNIE) which is a set of modules comprising a tokenizer, a gazetteer, a sentence splitter, a part of speech tagger, a named entities transducer, and a co-reference tagger. GATE has plugins for machine learning with Weka, RASP, MAXENT, SVM light, and a fast LibSVM integration, a perceptron implementation for managing ontologies like WordNet, plugins to query search engines like Google or Yahoo, and plugins for PoS tagging with Brill or TreeTagger. GATE also features a transducer to manipulate annotation on text called JAPE. GATE Developer is a GUI tool that integrates all GATE plugins and features. It provides a GUI interface to annotate documents and corpora.

GATE Mimir provides support for indexing and searching the linguistic and semantic information generated by GATE and facilitates querying the information using combinations of text, structural information, and SPARQL (Fig. 7).

4 QA Evaluation Metrics

Evaluation is very important for any IR system and question answering is no exception. Good evaluation metrics guide research and provide objective means of comparison between different approaches (Table 3).

Question answering evaluation metrics, on the other hand, can be applied across different languages. There are many evaluation metrics used with question answering; however, covering all these metrics is not in the scope of this article. We will only define the most used metrics in Arabic QA which are: precision, recall, accuracy, mean reciprocal rank (MRR), and C@1.

- $$\text{Accuracy}: \text{Acc} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{fp} + \text{tn} + \text{fn}} \tag{1}$$

- $$\text{Precision}: \text{P} = \frac{\text{tp}}{\text{tp} + \text{fp}} \tag{2}$$

- $$\text{Recall}: \text{R} = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{3}$$

- $$F\text{-measure}: \begin{array}{l} F = \frac{(1+\beta^2)PR}{(\beta^2 P)+R} \\ F\beta = 1 = \frac{2PR}{P+R} \end{array} \tag{4}$$

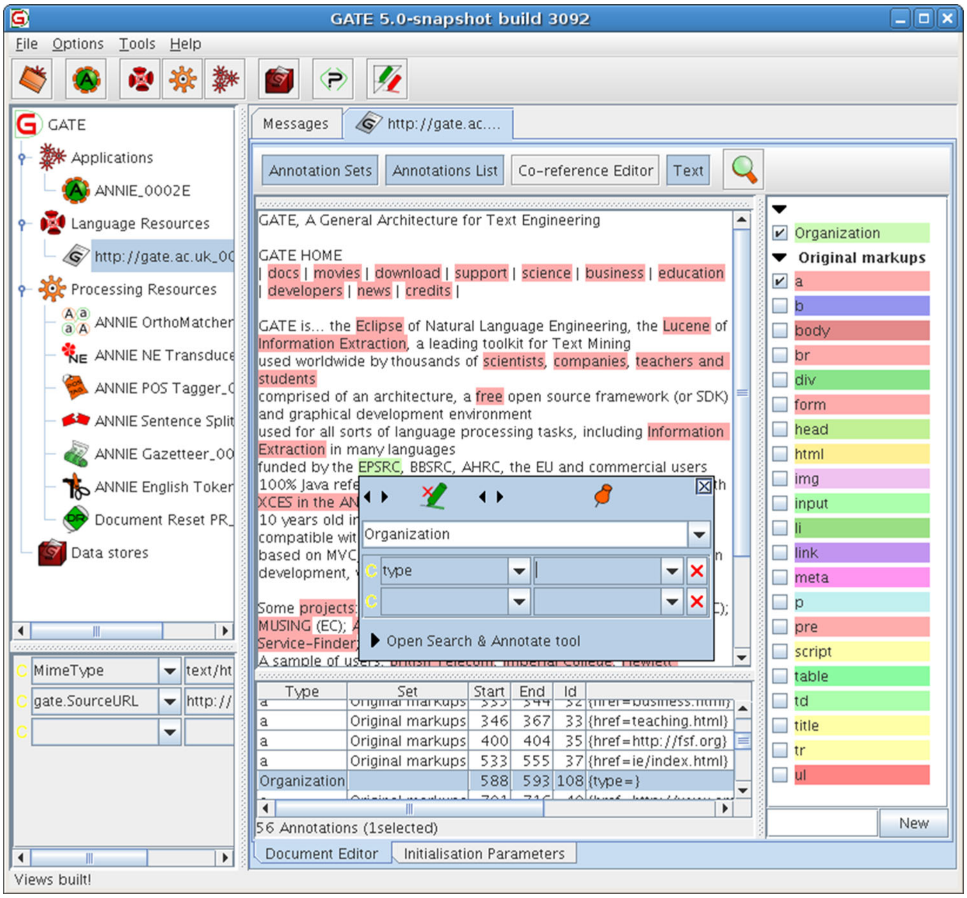
As illustrated in the previous contingency table and equations, accuracy is the number of relevant items retrieved and the number of not relevant items that are not retrieved divided by the number of all items. Precision is the number of relevant items that are retrieved divided by the number of all retrieved items. Recall is the number of relevant items that are retrieved divided by the number of all relevant items. Precision and recall complete each other because any IR system can maximize only one of them to 100 % easily. For example, if an IR system returns no results, then precision will be 100 % and if the system returns all documents even if they are not relevant, recall will be 100 %. Thus, it is incorrect to use only one of them [45]. F measure is a single metric that trades off precision versus recall. It is the weighted harmonic mean of precision and recall, where weight is denoted by a variable β. The default balanced F measure where $\beta = 1$ is commonly written as $F1$, which is short for $F_{\beta=1}$. See Eq. 4 [45].

Mean reciprocal rank (MRR) was introduced in TREC 2001 question answering track. If—for example—each question yields 5 possible answers. Each question is assigned a score equal to the reciprocal of the rank of the first correct answer. For example, if a question's first correct answer is in the 2nd place, it will receive a reciprocal rank (RR) of $1/2 = 0.5$. If the correct answer appears in the 5th place then the reciprocal rank is $1/5 = 0.2$. If the correct answer is the first one, then the RR will be $1/1 = 1$. If no answer was found in the returned five answers then the RR is 0. So the RR of a question can be one of these six values (0, 0.2, 0.25, 0.33, 0.5, 1). MRR is the mean of all the questions' reciprocal ranks. See Eq. 5. So this metric gives partial credit for answering a question but not in the first place, which is more realistic for QA systems [64].

- $$\text{Mean reciprocal rank}: \text{MRR} = \frac{\sum_{i=1}^{N} \frac{1}{\text{rank}_i}}{N} \tag{5}$$

where N is the number of questions. rank_i is the rank/order of the first correct answer of question i.

[9] GATE Official Website: http://gate.ac.uk.



MimeType text/ht
gate.SourceURL http://

GATE, A General Architecture for Text Engineering

GATE HOME
| docs | movies | download | support | science | business | education
| developers | news | credits |

GATE is... the Eclipse of Natural Language Engineering, the Lucene of
Information Extraction, a leading toolkit for Text Mining
used worldwide by thousands of scientists, companies, teachers and
students
comprised of an architecture, a free open source framework (or SDK)
and graphical development environment
used for all sorts of language processing tasks, including Information
Extraction in many languages
funded by the EPSRC, BBSRC, AHRC, the EU and commercial users
100% Java refe
XCES in the AN
10 years old in
compatible wit
based on MVC
development,

Some projects:
MUSING (EC); A
Service-Finder
A sample of users: British Telecom, Imperial College, Hewlett

☑ Organization
☑ a

Organization
type

Type	Set	Start	End	Id	
a	Original markups	333	344	32	{href=business.html}
a	Original markups	346	367	33	{href=teaching.html}
a	Original markups	400	404	35	{href=http://fsf.org}
a	Original markups	533	555	37	{href=ie/index.html}
Organization		588	593	108	{type=}

56 Annotations (1selected)

Fig. 7 The annotation editor window in GATE Developer

Table 3 Information retrieval contingency table

	Relevant	Not relevant
Retrieved	True positives (tp)	False positives (fp)
Not retrieved	False negative (fn)	True negative (tn)

C@1 is a metric introduced by Penas et al. [53] in question answering for machine reading evaluation (QA4MRE) at CLEF 2011 to encourage systems to leave some questions unanswered to reduce the amount of incorrect answers [53].

- $$C@1 := \frac{1}{n}\left(n_R + n_U\frac{n_R}{n}\right) \quad (6)$$

where: n_R: number of correctly answered questions n_U: number of unanswered questions n: total number of questions.

5 Arabic QA Test-Sets

Question answering evaluation in English and other Latin-based languages received greater interest over the past decade where conferences like TREC and CLEF contributed to building an ecosystem of test-sets and metrics to foster research in this area.

Covering English or Latin-based Languages QA test-sets is not in the scope of this survey but it is important to point out one important example of these test-sets to highlight the structure of a typical QA test-set. This example is the TREC (Text REtrieval Conference) 2001 QA test-set which was a set of 500 closed-class questions and a 3GB target corpus. Systems using this test-set were required to return 5 ranked answers for each question and the maximum length of each answer should be 50 bytes. Some of the questions did not have answer and the system should return the string "NIL" in this case. The target corpus was the AQUAINT Corpus of English News Text (LDC catalog number LDC2002T31), which consisted of documents from three sources: the AP newswire from 1998–2000, the New York Times newswire from 1998–2000, and the English portion of the Xinhua News Agency from 1996–2000. It contained about 1,033,000 documents with the size of 3 GB [65]. There are very few Arabic question answering test-sets and even fewer are available in the public domain. At the time of this writing

Benajiba et al. [16] created a test-set for their system "ArabiQA", which is composed of 200 questions with their answers and a corpus of 11,000 documents from the Arabic Wikipedia in SGML format (the format adopted in the CLEF and also the one accepted by the JIRS system). The test-set is publicly available on the Internet.[10] The proportion of each type of questions in the test-set is the same proportion adopted in Conference and Labs of the Evaluation Forum[11] (CLEF). This test-set also provides a redundancy (average of the number of passages returned for a question) of 3.28 using a light stemmer on all of its components [16].

Another test-set is "Arabic definition question answering" (ADQA) Corpus, which was created by Trigui et al. [62]. It is taken from the famous Arabic TV show "Who's gonna be a millionaire?" The test-set consists of the following:

- ArabicListDefQuest: a list of 50 Arabic organization definition questions.
- ArabicCorpusWikipedia: 50 files containing snippets collected from Wikipedia for the questions in ArabicListDefQuest.
- ArabicCorpusGoogle: 50 files containing snippets collected from Google for the questions in ArabicListDefQuest.
- ArabicListDefAnsw from—Google + Wikipedia—: 50 files containing the answers extracted from both Google and Wikipedia snippets for the questions in ArabicListDefQuest.
- ArabicListDefAnsw from—Google—: 50 files containing the answers extracted from both Google snippets for the questions in ArabicListDefQuest [62].

This test-set is also publicly available.[12]

Abouenour [6] introduced another test-set by manually translating the collection of 2,264 TREC and CLEF questions into Arabic. It is also publicly available.[13]

Another test-set is the test-set created by CLEF 2012 for the question answering for Machine Reading Evaluation (QA4MRE). QA4MRE at CLEF 2012 is the fourth campaign of its kind which is considered a new way of evaluating question answering [54]. Arabic QA was introduced for the first time in CLEF 2012. The 2012 test-set is composed of four topics: (1) "AIDS", (2) "climate change", (3) "music and society", and (4) "Alzheimer". Each topic has four reading tests. Every reading test has ten questions. Each of these questions has five answer options:

- 16 test documents (4 documents for each of the 4 topics)
- 160 questions (10 questions for each document)
- 800 answer choices/options (5 for each question)

Questions are designed to focus on testing the comprehension of one document only. These questions test the reasoning capabilities of the participating QA systems which may include inferences, relative clauses, elliptic expressions, meronymy, metonymy, temporal and spatial reasoning, and reasoning on quantities. These questions may also need some background knowledge that is not present in the test document. In such cases, Background Collections are provided by CLEF to fill this need. The questions types are

i Factoid: (where, when, by-whom)
ii Causal: (what was the cause/result of event X?)
iii Method: (how did X do Y? or in what way did X come about?)
iv Purpose: (why was X brought about? or what was the reason for doing X?)
v Which is true: (what can a 14 year old girl do?)

Questions are also classified according to their information needs as follows:

i 75 questions do not need extra knowledge (from background collections)
ii 46 questions need background knowledge
iii 21 questions need inference
iv 20 questions need information to be gathered from different sentences or paragraphs

6 QA Subtasks

Arabic question answering as a task is made up of three distinct subtasks: question analysis, Document/passage retrieval, and answer extraction. In this section, we will show how each subtask is implemented and the variations among different systems and implementations of Arabic QA in implementing these subtasks. See Fig. 8.

As illustrated in Fig. 8, question answering consists of 3 main phases: question analysis, passage retrieval, and answer extraction. In the question analysis phase, the expected answer type is determined according to the question words, and the question is formulated into a query to be ready for passage retrieval. In the passage retrieval phase, documents are separated into passages, and query formulated in the question analysis phase is used to search for the most relevant passages and rank them according to relevance. In the answer extraction phase, the retrieved passages are reranked

[10] Yassine Benajiba downloads: http://www1.ccls.columbia.edu/~ybenajiba/downloads.html.

[11] CLEF: http://www.clef-initiative.eu.

[12] Arabic Natural Language Research Group (ANLP-RG) Corpora: https://sites.google.com/site/anlprg/outils-et-corpus-realises.

[13] Sibaway Arabic question answering Area: http://sibawayh.emi.ac.ma/ArabicQA/m3.html.

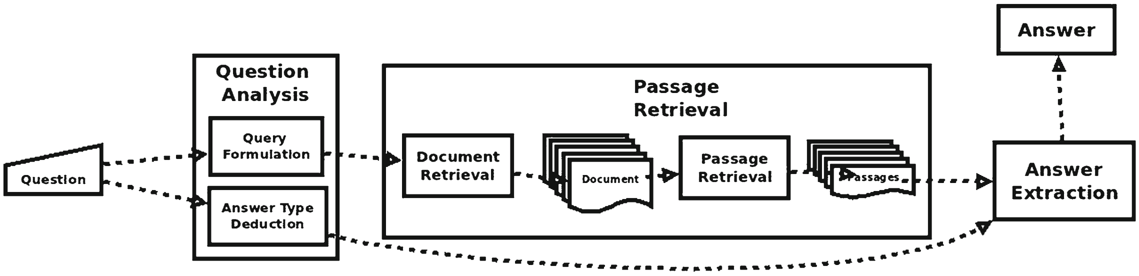


Fig. 8 Arabic question answering subtasks

according to the expected answer type detected in the question analysis phase, and the system responds with the first ranked answer.

In Table 4, a comparison is being made between the different Arabic question answering systems and the state-of-the-art English question answering system. Table 5 makes a feature to feature comparison between these systems and the next three subsections list their details.

6.1 Question Analysis

In an attempt to perform a better question analysis, Hammo et al. [34] parsed the question to extract its category and the type of answer required whether it is a name, a place, a quantity or a date, which makes it easier later in the answer extraction phase to select the right answer.

Rosso et al. [56] experimented with cross-language IR to answer Arabic questions from English documents. To analyze the question, they translated it and then made five different formulations to the question by verb and noun movement. They found out that the best results came out from verb reformulation in the translated question. However, the results were not promising as the precision decreased by about 20 % due to the ambiguity that translation adds to the question [56].

Rosso et al. [57] analyzed Arabic questions by eliminating stop words, extracting named entities and classified the questions into Name, Date, Quantity, and Definition questions according to the question word used. See Table 4.

Brini et al. [24] made some query formulation and extracted the expected answer type, question focus, and important question keywords. The question focus is the main noun phrase of the question that the user wants to ask about. For example, if the user's query is "What is the capital of Tunis?" then the question focus is "Tunis" and the keywords are "capital" and the expected answer type is a named entity for a location. Unfortunately, this work had only 100 questions which made it biased and unable to generalize.

Kanaan et al. 2009 [38] made four steps to analyze the question. They tokenized the question, then determined its type, then determined its focus which is the proper noun phrase and extracted the root of each non-stop word in the question. See Table 4.

Abdelbaki and Shaheen [1] analyzed the question by

(a) Tokenization and normalization

- Replacing initial إ , آ , أ by ا and the letter ئ by the sequence ىء
- Replacing final ى by ي and replace final ة by ه

(b) Determining answer type by question words (who, when...)
(c) Named Entity Recognition (gazetteer, maxent model)
(d) Focus determination by extracting the main NE
(e) Keywords Extraction by removing stop words using the Khoja stop list, which has 168 words and the 1,131 words translated from English.
(f) Keywords Expansion using the Arabic dictionary of synonyms. NEs are not expanded to avoid ambiguity.
(g) Stemming by Khoja's Stemmer and NEs are not stemmed
(h) Query generation of keywords into a Boolean formula [1]. See Table 4.

Bekhti et al. [14] segmented the question into interrogative noun, question's verb, and question's keywords. See Table 4.

6.2 Passage Retrieval

Awadallah and Rauber [13] experimented with Arabic and English QA and introduced two techniques to rank retrieved passages to select the best answer. The first technique is Answer and Question words Count (AQC) which is based on the number of questions and/or answer choice keywords occurring in result snippets. The second technique is Answer and Question words Association (AQA) which is the co-occurrence of question and answer choice keywords within the same result snippet's context. In other words, if there is a question with 5 candidate answers, then each candidate

Table 4 Comparison between Arabic QA systems and the state-of-the-art English QA system

System	Approach	Performance
AQAS	Question posed Arabic natural language	Not reported
	Used a knowledge-based model	
Mohammed et al. [48]	Search in structured data	
QARAB	Searched in Al-Raya newspaper corpus	Precision: 97.3 %
Hammo et al. [33,34]	Passage retrieval (PR) based on Salton's vector space model	Biased because:
		113 factoid questions only
	Treats a document as a "bag of words"	Questions posed by system creators
	Lexicon-based stemmer	
	PoS tagging and NER using [8]	Corpus and questions used are not publicly available
	Stop words removal	
	Expected answer type determination	Precision of 97.3 % is a lot greater than English state-of-the-art QA system in 2007 which is 70.6 % [49]
	Extract the named entity (NE) as the answer	
Rosso et al. [56]	Experimented with cross-language IR to answer Arabic questions from English documents	Precision of 39.5 % and MRR 0.31 by using English questions searching in English documents
	Translated questions then made 5 different formulations to it by verb and noun movement	Precision of 10.7 % and MRR 0.08 by using Arabic questions searching in English documents
		Decrease in precision by about 20 % due to the ambiguity incurred by translation
		Best results came out from verb reformulation in the translated question
Awadallah and Rauber [13]	Ranked passages according to answer and question words count (AQC) and answer and question words association (AQA)	Accuracy of 55 %
	Question of the famous Arabic TV show "Who's gonna be a millionaire?"	
ArabiQA	Followed CLEF guidelines to create their corpus	Answer extraction module precision of 83.3 %
	Removed stop words	Drawbacks:
	Extracted NEs	The used test-set was manually created and its creation details were not provided
Rosso et al. [57]	Classified the questions into name, date, quantity, and definition questions according to the question words	The system was not tested in an open domain
Benajiba et al. [16,17,19]	Assigning a higher rank for the passages that have a smaller distance between keywords: distance density model	
	Answer extraction task:	
	Tagged NEs in retrieved passages	
	Selected answers with expected type NEs	
	Patterns to select the final list of answers	
	Created a test-set to evaluate the answer extraction module in separation from the rest of the system	
QASAL	Query formulation	Recall equal to 100 % and precision equal to 94 %
Brini et al. [23,24]	Used NooJ local grammars	
	Used Google as a PR for the definition questions	Drawback: test-set size is too small, only 100 Factoid questions and 43 definition questions
	Extracted expected answer type, question focus and important question keywords	

Table 4 continued

System	Approach	Performance
Kanaan et al. [38]	Question analysis by tokenization, question type determination, determined its focus and extracted the root of each non-stop word in the question	Precision of IR system: 43 %
	PR based on Salton's vector space model	Drawback: test-set contained only 25 documents gathered from the Internet, 12 queries
DefArabicQA	Definition questions	MRR: 0.7
Trigui et al. [62]	Identified the candidate definitions using manual lexical patterns of sequence of words, letters and punctuation symbols	54 % of the questions were answered by the first candidate answer returned
	Used heuristic rules that they deduced from observing the form of some correct and incorrect definitions	Drawback: 50 organization definition questions only and the answers were assessed by only one Arabic native speaker
	Ranked candidate definitions, they ranked them according to the weight of the definition pattern, Snippet position, and the sum of word frequencies in the candidate definition	
Abouenour et al. [4–6]	Used translated CLEF and TREC questions	Accuracy: 20.20 %
	Used Yahoo search engine and JIRS passage retrieval system	Note: number of passages in JIRS was less than 1,000 which did not enable structure-based techniques to have great effect on the results
	Morphological and semantic query expansion using the Arabic WordNet	
	Enriched AWN to help query expansion	
	Ranking the passages based on distance density n-gram model	
	Used Amine platform to score and rank the retrieved passages semantically using concept graphs	
AQuASys	Segmented the question into interrogative noun, question's verb, and keywords	Recall: 97.5 %
Bekhti et al. [14]	Did not use a NER system	Precision: 66.25 %
		Drawback: cannot be used as it is on an untagged corpus as it used ANERcorp 316 documents of 150,000 tagged words and 80 questions only
Abdelbaki and Shaheen [1]	Analyzed the question by:	Accuracy: 86.25 %
	Tokenization and normalization and NER	Mean reciprocal rank (MRR): 0.87
	Determining expected answer type and question focus	Average response time: 2,262 ms on a machine with low specs
	Keywords extraction and expansion	(CPU: Intel® 1.60 GHz, RAM: 512 MB)
	Stemming by Khoja's Stemmer	
	Used semantic similarity between the question's focus and the candidate answer and made matching using N-grams	Drawback: Used ANERCorp for both training the NER module and as a document corpus which makes the results biased due to over-fitting
	Validated the answers using accuracy scoring and ranking	
	Used the ANERCorp 316 articles as a QA corpus and 240 questions	
State-of-the-art in English QA	Used TREC 2007 questions	Factoid QA accuracy: 70.6 %
Lymba's Power Answer 4	175 GB collection of blog entries and 2.5 GB newswire articles	List QA accuracy: 47.9 %
	Integrated semantic relations, advanced inference abilities, syntactically constrained lexical chains, and temporal contexts	
	Used strategies to answer each class of questions	
Moldovan et al. [49]	Each strategy has the 3 components of (1) question processing (2) passage retrieval (3) answer processing	
	Resolved fuzzy temporal expressions	
	Integrated With A Syntactic Parser, An Ner, A Semantic Parser, Ontologies, And A Logic Prover For Textual Inference In Answer Selection	
	Used Concept Tagger To Detect Event–Event Relations	

Table 5 Feature to feature comparison between Lymba's Power answer 4 and Arabic QA systems

	Test-set	Tokenization	Normalization	Stop-words removal	Question/answer patterns	Ontologies
AQAS Mohammed et al. [48]	Radiation diseases	Yes	–	Yes	Yes	–
QARAB Hammo et al. [33,34]	Al-Raya newspaper corpus and Questions put by users	Yes	–	Yes	–	–
Rosso et al. [56]	Translated CLEF-2003 Query Corpus	Yes	–	Yes	–	–
Awadallah and Rauber [13]	Arabic TV show "Who's gonna be a millionaire?" and TREC-2002 data	Yes	Yes	Yes	–	–
ArabiQA Rosso et al. [57] Benajiba et al. [16,17,19]	Followed CLEF guidelines to create their corpus	Yes	Yes	Yes	Yes	–
QASAL Brini et al. [23,24]	Extracted from Tunisian books for basic education and 100 questions	Yes	–	Yes	Yes	–
Kanaan et al. [38]	25 Internet documents and 12 questions	Yes	–	Yes	–	–
DefArabicQA Trigui et al. [62]	50 organization definition questions	Yes	–	Yes	Yes: answer patterns for definitions	–
Abouenour et al. [4–6]	Translated CLEF and TREC questions	Yes	Yes	Yes	–	Arabic WordNet
AquASys Bekhti et al. [14]	ANERcorp: 150,000 tagged tokens and 80 questions	Yes	–	Yes	Yes	–
Abdelbaki and Shaheen [1]	ANERcorp 316 articles and 240 questions	Yes	Yes	Yes	–	–
State-of-the-art in English QA Lymba's Power answer 4 Moldovan et al. [49]	TREC 2007 questions: 175 GB collection of blog entries & 2.5 GB newswire articles	Yes	–	Yes	Yes	Yes

	Stemming	PoS tagging	NER	Extract question focus	Semantic expansion	Knowledge base
AQAS Mohammed et al. [48]	Lexicon-based Stemmer	–	–	–	–	Yes
QARAB Hammo et al. [33,34]	Lexicon-based Stemmer	Yes [8]	Yes [8]	–	–	–
Rosso et al. [56]	–	–	–	–	–	–

Table 5 continued

	Stemming	PoS tagging	NER	Extract question focus	Semantic expansion	Knowledge base
Awadallah and Rauber [13]	Yes	–	–	–	–	–
ArabiQA Rosso et al. [57] Benajiba et al. [16,17,19]	Yes	Yes	Yes	–	–	–
QASAL Brini et al. [23,24]	Yes	Yes	Yes	Yes	–	–
Kanaan et al. [38]	Yes	Yes	Yes	Yes	–	–
DefArabicQA Trigui et al. [62]	–	–	–	–	–	–
Abouenour et al. [4–6]	Buckwalter morphological analyzer	–	Yes	–	Arabic WordNet	–
AquASys Bekhti et al. [14]	Khoja’s Stemmer	–	–	–	Yes	–
Abdelbaki and Shaheen [1]	Khoja’s Stemmer	–	Yes	Yes	Yes	–
State-of-the-art in English QA Lymba’s Power answer 4 Moldovan et al. [49]	Yes	Yes	Yes	Yes	–	Concept graphs and ontologies

	Answer type determination	PR module	Answer validation	Answer extraction	Other tools and techniques
AQAS Mohammed et al. [48]	Yes	–	–	Yes	–
QARAB Hammo et al. [33,34]	Yes	Based on Salton’s vector space model	–	Extracted the named entity to be the answer	–
Rosso et al. [56]	–	Yes	–	–	–
Awadallah and Rauber [13]	–	Google search engine	–	Ranked passages according to AQC & AQA see Table 4	–
ArabiQA Rosso et al. [57] Benajiba et al. [16,17,19]	Yes	JIRS	–	Yes: based on NER and Patterns	–
QASAL Brini et al. [23,24]	Yes	Google Search Engine	–	Yes: based on named entities	NooJ local grammars
Kanaan et al. [38]	Yes	Based on Salton’s vector space model	–	Yes: based on NER	–
DefArabicQA Trigui et al. [62]	–	Google and Wikipedia	–	Yes: Ranked answers with definition patterns and words frequencies	–

Table 5 continued

	Answer type determination	PR module	Answer validation	Answer extraction	Other tools and techniques
Abouenour et al. [4–6]	Yes	Yahoo search engine and JIRS	–	Yes	Amine platform concept graphs
AquASys Bekhti et al. [14]	Yes	Custom	–	Yes	–
Abdelbaki and Shaheen [1]	Yes	Uses a simple keyword-based IR system	Yes: Using accuracy scoring and ranking	Yes: using Semantic & N-gram similarity	–
State-of-the-art in English QA Lymba's Power answer 4	Yes	Apache Lucene	Yes: using optimal threshold learned from past TREC evaluations	Yes: using NER and concept tagger	Concept tagger to detect event–event relations, logic prover, syntactic parser, semantic parser, ontologies
Moldovan et al. [49]					Semantic relations, inference, syntactically constrained lexical chains, Temporal contexts, Resolving Fuzzy temporal expressions, and using strategies to answer each class of questions

answer is joined with the question and passed to the passage retrieval module. A retrieved passage is then assigned a higher ranking if it contains more question and candidate answer keywords (AQC). If the candidate answer and question keywords appear nearer to each other in the retrieved passage it is also assigned a higher ranking (AQA). They held their experiments on the question of the famous Arabic TV show "Who's gonna be a millionaire?" and TREC-2002 QA track questions. Their experiments revealed an average performance of 55–62 %. The AQA strategy had better performance on the Arabic language questions while AQC was better for English language tasks. This may be due to the morphological complexity of Arabic that resulted in retrieving only precise phrases if they exist, rather than retrieving split segments [13].

Benajiba et al. [19] ranked the retrieved passages according to the relevant question terms appearing in the passage and assigned a higher rank for the passages that have a smaller distance between keywords which is called the Distance Density model. See Fig. 9.

Kanaan et al. [38] used a passage retrieval system following Salton's vector space model using query words' weight, and cosine similarity between documents' words and question words. Their system tokenized every document, removed the stop words, and carried out root extraction and term weighting. However, their test-set was only 25 documents gathered from the Internet, 12 queries (questions), and some relevant documents provided by themselves. See Table 4.

Abouenour et al. [4–6] explained an enhanced passage retrieval built on the JIRS passage retrieval system. He followed a three-level approach in his passage retrieval system:

(a) Keyword-based level: morphological and semantic query expansion using the Arabic WordNet including the concept hypernyms, hyponyms, synonyms, and definition.
(b) Structure-based level: ranking the passages based on Distance Density n-gram Model giving higher rank to passages that have the question words appear nearer one another.
(c) Semantic Reasoning level: where he used Amine Platform to score and rerank the retrieved passages semantically using concept graphs to find the most relevant answer passage.

However, the number of processed passages in JIRS was less than 1000 which did not enable structure-based techniques to have great effect. See Table 4.

The CLEF 2012 campaign had 2 Arabic QA attempts. The first attempt is IDRAAQ by Abouenour et al. Its NER is achieved by mapping the YAGO[14] ontology and Arabic WordNet. The passage retrieval module of IDRAAQ is based on two levels:

[14] Yet Another great ontology: http://www.mpi-inf.mpg.de/YAGO-naga/YAGO/downloads.html.

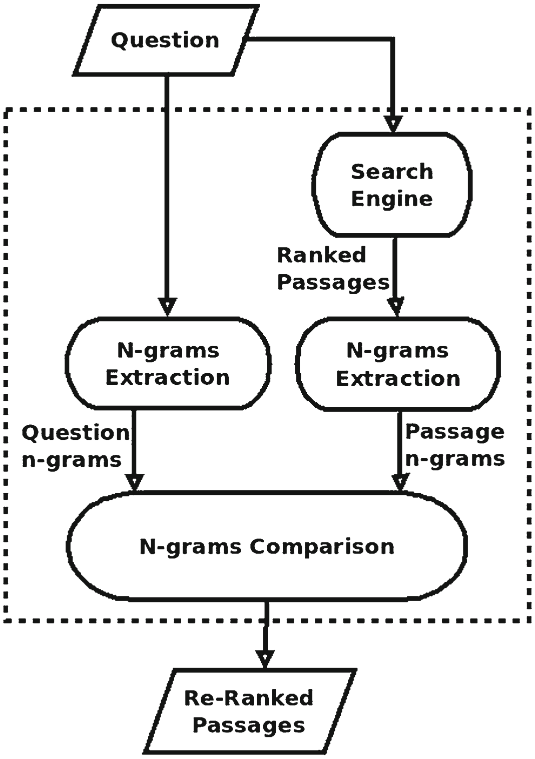


Fig. 9 JIRS passage retrieval system architecture [19]

1. Keyword-based level: based on Query Expansion process relying on Arabic WordNet semantic relations
2. Structure-based level: based on a Distance Density N-gram Model-based passage retrieval system which is JIRS

IDRAAQ [7] has reached a very low but encouraging accuracy of 0.13 as it did not use any CLEF background collections. It is also considered the best Arabic QA system because it scored 0.21 in the c@1 metric which means that it marked some of the questions as unanswered [7].

The other attempt is the QA system created by Trigui et al. [63]. They determined question focus and searched for the focus in the test passages, and the collected passages are aligned with the multiple answer choices of the question. If there is no answer included in passages, a list of pairs of words is generated from the background collection according to a list of inference rules. Any word from the answer option that does not exist in the passages is replaced by its inference word. If there is no answer included in the passages after this step then the question isconsidered unanswered. The accuracy and C@1 of this attempt is 0.19 which means that there are no questions marked as unanswered [63].

However, both attempts were not as successful as the best system in the English language at CLEF 2012 by Bhaskar et al. [21], which has the accuracy of 0.53 and the c@1 of 0.65. This may pertain to their different approach to the problem as they combined each of the 5 answer choices with the question in a hypothesis then searched for the keywords of each hypothesis in the passages and ranked the passages according textual entailment [21]. Table 6 and the chart in Fig. 10 compare the two Arabic QA systems in QA4MRE at CLEF 2012 and the state-of-the-art English QA in the same conference. Table 7 illustrates the techniques deployed in the question analysis and Answer Validation modules on the two state-of-the-art Arabic and English QA4MRE systems at CLEF 2012.

However, it is worth mentioning that some of the reading-test documents and questions have translation errors, as reported by Abouenour et al. in IDRAAQ, which could have impacted the performance of both IDRAAQ and Trigui et al. attempts. For example, in reading-test 4 question 4 the translation of

“What is the mechanism by which HIV-positive Brazilians receive free ARV drugs?” is

" ماهي الأليات المستعملة لإعطاء البرازيليين المصابين بداء نقصان المناعة البشرية المضادة
للفيروسات القهقرية مجاناً؟" ,

which is not considered comprehensible in Arabic.

6.3 Answer Extraction and Validation

Trigui et al. [62] tackled the definition type of questions. They first identified the candidate definitions using manual lexical patterns of sequence of words, letters, and punctuation symbols. Then they used some heuristic rules that they deduced from observing the form of some correct and incorrect definitions. After they extracted the candidate definitions, they ranked them according to three criteria which are (i) pattern weight of the pattern that matched the candidate definition, (ii) snippet position of the snippet that contains the candidate definition in the snippets collection, and (iii) the sum of word frequencies in the candidate definition. However, their evaluation was not good enough as they tested on 50 organization definition questions only and the answers were assessed by only one Arabic native speaker. See Table 4.

Abdelbaki and Shaheen [1] used semantic similarity between the question’s focus and the candidate answer and made matching using n-grams. After that they validated the answers using accuracy scoring and ranking. The results

Table 6 Comparison between Arabic QA4MRE @ CLEF 2012 and English state-of-the-art system

QA4MRE system	Deployed components	Performance
IDRAAQ Abouenour et al. [7]	NER by mapping the YAGO ontology and Arabic WordNet. Did not use CLEF background collections—PR based on query expansion using AWN Semantic relations, and distance density N-gram model of JIRS	C@1 : 0.21 Accuracy: 0.13
Trigui et al. [63]	Determine question focus PR retrieved passages are aligned with the multiple answer choices of the question Semantic expansion using inference rules on the background collection	C@1: 0.19 Accuracy: 0.19
State-of-the-art English QA4MRE @ CLEF 2012 Bhaskar et al. [21]	Combined each answer choice with the question in a hypothesis PR searched for hypothesis keywords Ranked passages according textual entailment	C@1 : 0.65 Accuracy: 0.53

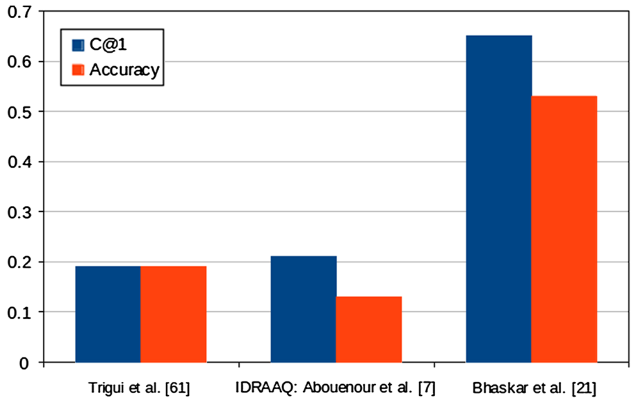


Fig. 10 Performance of QA4MRE systems @ CLEF 2012

they achieved were 86.25 % accuracy and an mean reciprocal rank (MRR) of 0.87. They also provided the average response time which was 2,262 ms on a machine with low specs (CPU: Intel® 1.60 GHz, RAM: 512 MB) [1]. However this work used the ANERCorp 316 articles as a QA corpus and posed 240 questions on this small corpus which makes the redundancy passages not enough to test the passage retrieval and the answer extraction modules. It is also noticed that by using ANERCorp corpus for training the Arabic named entity recognition (NER) classifier then using it as the QA corpus will make the system over-fitted for this corpus and may not reach the same results on other unseen texts.

Benajiba et al. [16], in their system named ArabiQA, approached the answer extraction task in three steps: (see Fig. 11).

(a) Using an NER system to tag all NEs in the retrieved passages.
(b) Selecting candidate answers NEs that has the same expected answer type only.
(c) Applying a set of patterns to select the final list of answers.

Moreover, they created a test-set solely to evaluate their answer extraction module in separation from the rest of the system. This test-set was made up of four lists:

(a) List of the questions
(b) List containing the type of each question
(c) List of manually selected passages that contain the right answers for the questions
(d) List of correct answers

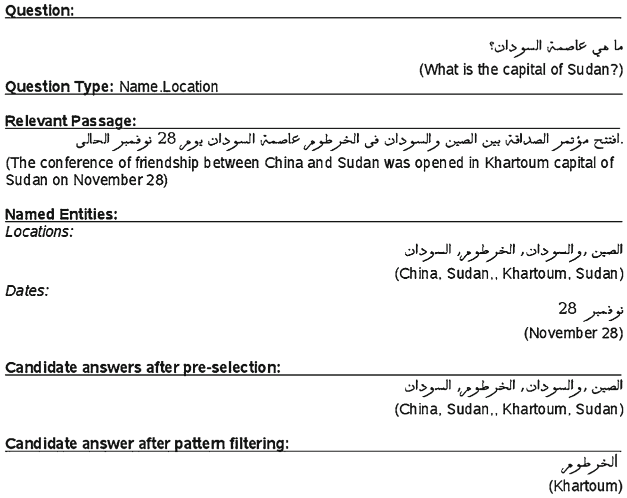
Their AE (answer extraction) module performed at a precision of 83.3 % where the precision here is calculated by dividing the number of correct answers over the number of questions [16].

7 Future Trends

As noticed from this survey, work in the field of Arabic QA is very limited which mandates a deep look in the future trends of this area. Most of the work in the field of Arabic QA is focused on open domain question answering while very few attempts approached restricted domain question answering. It is also noticed that there is almost no Arabic QA research done using theorem proving and deep reasoning. Among

Table 7 Techniques of the state-of-the-art QA4MRE Arabic and English systems @ CLEF 2012

Criterion	IDRAAQ: Abouenour et al. [7]	State-of-the-art English system Bhaskar et al. [21]
Question analysis and Linguistic processing methods		
Automatically acquired patterns	Yes	Yes
PoS tagging	Yes	Yes
n-grams	Yes	Yes
Chunking	Yes	Yes
Dependency analysis	Yes	Yes
NER	Yes	Yes
Temporal expressions	–	Yes
Numerical expressions	–	Yes
Syntactic transformations	–	Yes
Grammatical functions (subject, Object, etc.)	Yes	Yes
Semantic parsing	Yes	Yes
Semantic role labeling	Yes	Yes
Predefined sets of relations	Yes	–
Frames	Yes	–
Conceptual graphs	Yes	–
Similarity scoring	Yes	–
Answer validation techniques		
Redundancies in collection	Yes	–
Lexical similarity (term overlapping)	Yes	Yes
Syntactic similarity	Yes	Yes
Semantic similarity	Yes	Yes



Question:

ما هي عاصمة السودان؟

(What is the capital of Sudan?)

Question Type: Name.Location

Relevant Passage:

افتتح مؤتمر الصداقة بين الصين والسودان فى الخرطوم عاصمة السودان يوم 28 نوفمبر الحالى.

(The conference of friendship between China and Sudan was opened in Khartoum capital of Sudan on November 28)

Named Entities:

Locations:

الصين ,والسودان, الخرطوم, السودان

(China, Sudan,, Khartoum, Sudan)

Dates:

نوفمبر 28

(November 28)

Candidate answers after pre-selection:

الصين ,والسودان, الخرطوم, السودان

(China, Sudan,, Khartoum, Sudan)

Candidate answer after pattern filtering:

الخرطوم

(Khartoum)

Fig. 11 Illustrating example of the answer extraction module's performance steps [16]

the untapped areas in Arabic QA is using inference-based and logic-based approaches to QA. It is also important to give more attention to semantics as most research concentrated on the morphological and syntactic aspects of Arabic to approach the Arabic QA task.

7.1 More Research on Arabic Restricted Domain QA

Since the introduction of AI techniques, question answering in English and Latin-based languages was one of the most important AI problems. However, due to the size of information and limitations on processing power, most of the QA applications were of the restricted domain type. This changed after TREC introduced a new approach to question answering in 1999, offloading the QA problem to the information retrieval field of computing. Question answering was then reduced to the three aforementioned subtasks and great attention was given to the passage retrieval task as being an information retrieval task. This led to the interest in Open Domain QA by using the advances in the information retrieval field to push the question answering task. However, little work has been done on the semantic, logic, inference, and deep reasoning approaches to QA. It is worth mentioning that restricted domain QA may not benefit from redundancy-based approaches as most restricted domain corpora are small in size which does not allow redundancy to have great effect in selecting the right answer.

The first and almost only attempt for restricted domain Arabic question answering was AQAS, an Arabic QA system that was specialized in the restricted domain of radiation and its effects [48]. However, this system was more or less a natural language interface to a structured database of frames about this area. Almost all of the systems that followed after that were open domain QA systems that were either used to search a group of documents from a newspaper for example or used to search the Internet as an open domain. Research in restricted domain QA makes semantic tasks like word sense disambiguation easier. It is also very common to find domain rules affecting how the question is posed and how the answer is formulated. Minock defined how to choose a restricted domain, stating that a restricted system should be [47]:

- Circumscribed: which means that the domain should be bounded so that the user has previous expectation of what this domain should include. Agriculture, architectural engineering or any field of science is a circumscribed domain, but the domain of news and current events is not circumscribed as it does not have governing rules and constraints and the user will not be able to expect what the QA system knows to ask it.
- Complex: the domain should not be very easy to the extent that any simple application can do the required task. This

type of domains is for commercial applications not for research.
- Practical: which means that it is of great importance to many users, so that it will be used after being created.

Among the restricted domains that can be served by Arabic QA are

- Military Information: helps find answers to strategic questions from military Arabic documents.
- Judicial and Legal Information: helps judges and lawyers to get a simple answer for any question from the tons of legal documents available.
- Police Information: can help the Arab countries' police and internal affairs to take well-informed decisions that help in reducing crime and ensure public safety.

7.2 Use of Deep Application-Dependent Approaches

In restricted domain QA, it is more helpful to use application-dependent constraints and rules to guide the question analysis and answer extraction and validation. Depending on the available resources in a restricted domain can be of great advantage over open domain QA systems that only rely on common ontologies and WordNet. For example, AQAS, which was the first and only restricted domain Arabic QA system, made use of the deep application-dependent features of the domain of radiation. AQAS knowledge base frames had two types according the types of information in this knowledge base [48]:

- Object/Person frames: that describes physical features like size, shape and contents.
- Action Frames: that describe dynamic features like disease and radiation effect frames.

7.3 Intensive Use of Semantics

Al-Safadi et al. [9] developed a domain-dependent semantic-based search engine for Arabic blogs and proved that information retrieval precision is increased by combining Natural Language Processing with ontologies. They also mentioned that keyword-based approaches are prone to low retrieval precision for Arabic content if they are not combined with ontologies to enrich them semantically [9]. However, most of the research in the field of Arabic QA focused on morpho-syntactic approaches while very few used semantic approaches. Several research like the work of Abouenour et al. [4–6] used the Arabic WordNet and even contributed to the richness of this ontology. Yet, there is still a lot of research to be done in the field of word sense disambiguation, co-reference resolution, and ontology-based reasoning and their integration with Arabic question answering.

7.4 Use of Theorem Proving and Deep Reasoning

In IBM, a group of 20 researchers worked for 3 years to create Watson, a DeepQA system, and managed for the first time to make a computer system that beat the best players in Jeopardy a quiz TV show. This system was able to answer 85 % of the questions in 5 s to beat the human expert players. They used what they called DeepQA that used rule-based deep parsing and statistical classification methods to decompose the question into sub-questions [30]. IBM Watson used a special, very powerful hardware to accomplish this task, which means there is a lot to be done to scale this type of systems for the web or for commercial use. Some of these techniques were used not only with English but also with German where deep reasoning and theorem proving techniques were used to answer forum questions by a German QA system named Loganswer. Forum users do not require real-time answers which allowed some extra time for slow deep reasoning [52]. Until the time of this writing, these approaches were not used in Arabic QA.

7.5 Use of Logic-Based and Inference-Based Approaches

Mollá et al. [50], in the ExtrAns system, used logic-based approaches to answer UNIX manual questions. ExtrAns used linguistic information extracted from the documents and terminological knowledge about the UNIX domain. It transformed the sentences in the documents into semantic representations called Minimal Logical Forms (MLFs) and they stored these MLFs in a knowledge base. When the user poses a question, the same mechanism is applied to the question and the MLF of the question is proved by deduction over the MLFs of the document sentences in the Knowledge Base [50]. Kontos et al. [40] also used logic- and inference-based approaches in question answering in their system (AROMA) that used causal relationships to deduce the answer both by creating preprocessed formal semantic representations giving them to Prolog as an inference engine, and by inferring through the Natural Language text directly using a more complex inference engine. Logic and inference based approaches in Arabic QA have great potential, due to the shortage of research in this area so far [40].

8 Conclusion

In this survey, we reviewed the Arabic-specific difficulties and the tools created to tame these difficulties. The tools, we reviewed, include named entity recognition tools, passage retrieval tools, logic and inference tools, and morphological analysis toolkits for text normalization, tokenization, part-of-speech tagging, diacritization, base phrase chunking, stemming, and lemmatization. Arabic QA language resources, corpora, test-sets, and evaluation metrics were also reviewed. We also went through many attempts to solve the Arabic question answering problem by breaking it down into its NLP and IR subtasks. We reviewed the three main subtasks of Arabic QA (question analysis, passage retrieval, and answer extraction), and how researchers approached these subtasks. Finally, the Arabic QA future trends were highlighted to guide new research in this area.

References

1. Abdelbaki, H.; Shaheen, M.; Badawy, O.: ARQA high-performance arabic question answering system. In: Proceedings of Arabic Language Technology International Conference (ALTIC) (2011)
2. Abdelrahman, S.; Elarnaoty, M.; Magdy, M.; Fahmy, A.: Integrated machine learning techniques for Arabic named entity recognition. IJCSI 1 (2010)
3. Abouenour, L.; El Hassani, S.; Yazidy, T.; Bouzouba, K.; Hamdani, A.: Building an Arabic morphological analyzer as part of an open Arabic NLP platform. In: The Language Resources and Evaluation Conference (LREC), Marrakech, Morocco, 31st May (2008)
4. Abouenour, L.; Bouzoubaa, K.; Rosso, P.: Three-level approach for passage retrieval in Arabic question/answering systems. In: Proc. of the 3rd International Conference on Arabic Language Processing CITALA2009, Rabat, Morocco (2009)
5. Abouenour, L.; Bouzouba, K.; Rosso, P.: An Evaluated Semantic Query Expansion and Structure-Based Approach for Enhancing Arabic Question/Answering (2010)
6. Abouenour, L.: On the improvement of passage retrieval in arabic question/answering (Q/A) systems. Natural Lang. Process. Inf. Syst., pp. 336–341 (2011)
7. Abouenour, L.; Bouzoubaa, K.; Rosso, P.: IDRAAQ: new arabic question answering system based on query expansion and passage retrieval. In: CLEF 2012 Workshop on Question Answering For Machine Reading Evaluation (QA4MRE) (2012)
8. Abuleil, S.; Evens, M.: Discovering Lexical Information by Tagging Arabic Newspaper Text. Workshop on Semantic Language Processing. COLING-ACL '98, University of Montreal, Montreal, PQ, Canada, Aug. 16 1998, pp. 1–7 (1998)
9. Al-Safadi, L.; Al-Rgebh, D.; AlOhali, W.: A comparison between ontology-based and translation-based semantic search engines for Arabic blogs. Arab. J. Sci. Eng. **38**(11), 2985–2992 (2013)
10. Alshalabi, R.: Pattern-based Stemmer for finding Arabic roots. Inf. Technol. J. **4**(1), 38–43 (2005)
11. Attia, M.; Rashwan, M.; Ragheb, A.; Al-Badrashiny, M.; Al-Basoumy, H.; Abdou, S.: A compact Arabic lexical semantics language resource based on the theory of semantic fields. In: Advances in Natural Language Processing, pp. 65–76. Springer, Berlin, Heidelberg (2008)
12. Attia, M.; Rashwan, M.; Al-Badrashiny, M.A.S.A.A.: Fassieh, a semi-automatic visual interactive tool for morphological, PoS-Tags, phonetic, and semantic annotation of Arabic Text Corpora. In: IEEE Transactions on Audio, Speech, and Language Processing, vol. 17(5), pp. 916–925 (2009)
13. Awadallah, R.; Rauber, A.: Web-based multiple choice question answering for English and Arabic questions. Adv. Inf. Retr. 515–518 (2006)
14. Bekhti, S.; Rehman, A.; Al-Harbi, M.; Saba, T.: AQuASys an Arabic question-answering system based on extensive question analysis and answer relevance scoring. Inf. Comput. Int. J. Acad. Res. **3**(4), 45–54 (2011)
15. Benajiba, Y.; Rosso, P.: ANERsys 2.0: conquering the NER task for the Arabic language by combining the maximum entropy with PoS-tag information. In: Proc. of Workshop on Natural Language-Independent Engineering, IICAI-2007 (2007)
16. Benajiba, Y.; Rosso, P.; Lyhyaoui, A.: Implementation of the AraBiQA question answering system's components. In: Proc. Workshop on Arabic Natural Language Processing, 2nd Information Communication Technologies Int. Symposium, ICTIS-2007, Fez, Morroco, April, pp. 3–5 (2007)
17. Benajiba, Y.; Rosso, P.: Arabic question answering. Diploma of advanced studies. Technical University of Valencia, Spain (2007)
18. Benajiba, Y.; Rosso, P.; BenedíRuiz, J.: ANERsys: an Arabic named entity recognition system based on maximum entropy. Comput. Linguist. Intell. Text Process. 143–153 (2007)
19. Benajiba, Y.; Rosso, P.; Gómez Soriano, J.: Adapting the JIRS passage retrieval system to the Arabic language. Comput. Linguist. Intell. Text Process. 530–541 (2007)
20. Benajiba, Y.; Rosso, P.: Arabic named entity recognition using conditional random fields. In: Proc. of Workshop on HLT NLP within the Arabic World, LREC, vol. 8, pp. 143–153 (2008)
21. Bhaskar, P.; Pakray, P.; Banerjee, S.; Banerjee, S.; Bandyopadhyay, S.; Gelbukh, A.: Question answering system for QA4MRE@CLEF 2012. In: CLEF 2012 Workshop on Question Answering For Machine Reading Evaluation (QA4MRE) (2012)
22. Bouzouba, K.; Kabbaj, A.: An Integrated Development Platform for Arabic Language Processing. ISCAL-07.s (2007)
23. Brini, W.; Ellouze, M.; Trigui, O.; Mesfar, S.; Belguith, H.L.; Rosso, P.: Factoid and Definitional Arabic Question Answering System. Post-Proc. NOOJ-2009, Tozeur, Tunisia, June, 8–10 (2009)
24. Brini, W.; Ellouze, M.; Mesfar, S.; Belguith, L.H.: An Arabic question-answering system for factoid questions. In: IEEE International Conference on Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009, pp. 1–7 (2009)
25. Buckwalter, T.: Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium, catalog number LDC2002L49, ISBN 1-58563-257-0 (2002)
26. Buscaldi, D.; Gómez, J.M.; Rosso, P.; Sanchis, E.: The UPV at QA@ CLEF 2006. In: Working Notes for the CLEF 2006 Workshop (2006)
27. Diab, M.: Second generation AMIRA tools for Arabic processing: fast and robust tokenization, PoS tagging, and base phrase chunking. In: Proceedings of the second international conference on arabic language resources and tools, pp. 285–288 (2009)
28. Elghamry, K.; Al-Sabbagh, R.; El-Zeiny, N.: Cue-based bootstrapping of Arabic semantic features. JADT 2008: 9es Journées internationales d'Analyse statistique des Données Textuelles (2008)
29. Elkateb, S.; Black, W.; Vossen, P.; Farwell, D.; Rodríguez, H.; Pease, A.; Alkhalifa, M.: Arabic WordNet and the challenges of Arabic. In: Proceedings of Arabic NLP/MT Conference, London, UK (2006)
30. Ferrucci, D.; Brown, E.; Chu-Carroll, J.; Fan, J.; Gondek, D.; Kalyanpur, A.A.; Welty, C.; others.: Building Watson: an overview of the DeepQA project. AI Mag. **31**(3), 59–79 (2010)
31. Gomez, J.M.; Montes-Gomez, M.; Sanchis, E.; Villasenor-Pineda, L.; Rosso, P.: Language independent passage retrieval for question answering. In: Fourth Mexican International Conference on Artificial IntelligenceMICAI 2005, Lecture Notes in Computer Science, pp. 816–823, Monterrey, Mexico, 2005. Springer, Berlin (2005)
32. Habash, N., Rambow, O., Roth, R.: MADA+TOKAN: a toolkit for Arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. In: Proceedings of the

2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt, pp. 102–109 (2009)

33. Hammo, B.; Abu-Salem, H.; Lytinen, S.: QARAB: a question answering system to support the Arabic language. In: Proceedings of the ACL-02 workshop on computational approaches to semitic languages, pp. 1–11. Association for Computational Linguistics (2002)
34. Hammo, B.; Abuleil, S.; Lytinen, S.; Evens, M.: Experimenting with a question answering system for the Arabic language. Comput. Human. **38**(4), 397–415 (2004)
35. Harmanani, H.M.; Keirouz, W.T.; Raheel, S.: A rule-based extensible Stemmer for information retrieval with application to Arabic. Int. Arab. J. Inf. Technol. **3**(3), 265–272
36. Hatcher, E.; Gospodnetic, O.; McCandless, M.: Lucene in action (2004)
37. Kadri, Y.; Nie, J.Y.: Effective Stemming for Arabic information retrieval. In: Proceedings of the Challenge of Arabic for NLP/MT Conference, Londres, Royaume-Uni (2006)
38. Kanaan, G.; Hammouri, A.; Al-Shalabi, R.; Swalha, M.: A new question answering system for the Arabic language. Am. J. Appl. Sci. **6**(4), 797–805 (2009)
39. Khoja, S.; Garside, R.: Stemming Arabic text. Computing Department, Lancaster University, Lancaster, UK (1999)
40. Kontos, J.; Malagardi, I.O.A.N.N.A.; Peros, J.O.H.N.: Question answering and rhetoric analysis of biomedical texts in the aroma system. In: Proceedings of the 7th HERCMA: Hellenic European conference in computer mathematics and its applications, Athens, Greece (2005)
41. Larkey, L.S.; Connell, M.E.: Arabic Information Retrieval at UMass in TREC-10. Massachusetts Univ Amherst Center for Intelligent Information Retrieval (2006)
42. Larkey, L.S.; Ballesteros, L.; Connell, M.E.: Light stemming for Arabic information retrieval. In: Arabic Computational Morphology, pp. 221–243. Springer, Netherlands (2007)
43. Laurent, D.; Séguéla, P.; Nègre, S.: QA better than IR? In: Proceedings of the Workshop on Multilingual Question Answering, pp. 1–8. Association for Computational Linguistics (2006)
44. Maamouri, M.; Bies, A.; Buckwalter, T.; Mekki, W.: The Penn Arabic Treebank: building a large-scale annotated Arabic Corpus. In: NEMLAR Conference on Arabic Language Resources and Tools, pp. 102–109 (2004)
45. Manning, C.D.; Raghavan, P.; Schütze, H.: Introduction to information retrieval, vol. 1. Cambridge University Press, Cambridge (2008)
46. Mesfar, S.: Morpho-Syntactic Analysis and Automatic Recognition of Named Entities in Standard Arabic. University of Franche-account, Academic (2008)
47. Minock, M.: Where are the ‘killer applications’ of restricted domain question answering. In: Proceedings of the IJCAI Workshop on Knowledge Reasoning in Question Answering, p. 4 (2005)
48. Mohammed, F.A.; Nasser, K.; Harb, H.M.: A Knowledge Based Arabic Question Answering System (AQAS). ACM SIGART Bull. **4**(4), 21–30 (1993)
49. Moldovan, D.; Clark, C.; Bowden, M.: Lymba’s PowerAnswer 4 in TREC 2007. In: Proceedings of the Sixteenth Text REtrieval Conference (TREC 2007). Gaithersburg (2007)
50. Molla, D.; Schwitter, R.; Rinaldi, F.; Dowdall, J.; Hess, M.: Extrans: extracting answers from technical texts. IEEE Intell. Syst. **18**(4), 12–17 (2003)
51. O’Steen, D.; Breeden, D.: Named Entity Recognition in Arabic: A Combined Approach (2009)
52. Pelzer, B.; Glöckner, I.; Dong, T.: Loganswer in question answering Forums. In: 3rd International Conference on Agents and Artificial Intelligence (ICAART 2011), SciTePress, pp. 492–497 (2011)
53. Penas, A.; Rodrigo, A.; del Rosal, J.: A simple measure to assess non-response. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 1415–1424 (2011)
54. Penas, A.; Hovy, E.; Forner, P.; Rodrigo, A.; Sutcliffe, R.; Sporleder, C.; Forascu, C.; Benajiba, Y.; Osenova, P.: Overview of QA4MRE at CLEF 2012: question answering for machine reading evaluation. In: CLEF 2012 Workshop on Question Answering For Machine Reading Evaluation (QA4MRE) (2012)
55. Rashwan, M.A.; Al-Badrashiny, M.A.S.A.A.; Attia, M.; Abdou, S.M.; Rafea, A.: A stochastic Arabic diacritizer based on a hybrid of factorized and unfactorized textual features. IEEE Transactions on Audio Speech Lang. Process. **19**(1), 166–175 (2011)
56. Rosso, P.; Lyhyaoui, A.; Peñarrubia, J.; y Gómez, M.M.; Benajiba, Y.; Raissouni, N.: Arabic-English question answering. In: Proc. Symposium on Information Communication Technologies Int., Tetuan, Morocco (2005)
57. Rosso, P.; Benajiba, Y.; Lyhyaoui, A.: Towards an Arabic question answering system. In: Proc. 4th Conf. on Scientific Research Outlook Technology Development in the Arab world, SROIV, Damascus, Syria, pp. 11–14 (2006)
58. Sidrine, S.; Souteh, Y.; Bouzoubaa, K.; Loukili, T.: SAFAR: vers une Plateforme Ouverte pour le Traitement Automatique de la Langue Arabe. In: Proc of the 6th Intelligent Systems: Theory and Applications SITA 2010 Conference, Rabat, Morocco (2010)
59. Silberztein, M.: NooJ: a linguistic annotation system for corpus processing. In: Proceedings of HLT/EMNLP on Interactive Demonstrations, pp. 10–11. Association for Computational Linguistics (2005)
60. Smucker, M.D.; Allan, J.; Dachev, B.: Human question answering performance using an interactive information retrieval system. Center for Intelligent Information Retrieval Technical Report IR-655, University of Massachusetts (2008)
61. Taghva, K.; Elkhoury, R.; Coombs, J.: Arabic Stemming without a root dictionary. In: IEEE International Conference on Information Technology: Coding and Computing, 2005. ITCC 2005, vol. 1, pp. 152–157 (2005)
62. Trigui, O.; Belguith, H.L.; Rosso, P.: DefArabicQA: Arabic definition question answering system. In: Workshop on Language Resources and Human Language Technologies for Semitic Languages, 7th LREC, Valletta, Malta, pp. 40–45 (2010)
63. Trigui, O.; Belguith, L.H.; Rosso, P.; Amor, H.B.; Gafsaoui, B.: Arabic QA4MRE at CLEF 2012: Arabic question answering for machine reading evaluation. In: CLEF 2012 Workshop on Question Answering For Machine Reading Evaluation (QA4MRE) (2012)
64. Voorhees, E.M.: Question answering in TREC. In: Proceedings of the Tenth International Conference on Information and Knowledge Management, pp. 535–537. ACM, New York (2001)
65. Voorhees, E.M.; Harman, D.: Overview of TREC 2001. In: Proceedings of TREC, pp. 1–15 (2001)
66. Zaghouani, W.; Pouliquen, B.; Ebrahim, M.; Steinberger, R.: Adapting a resource-light highly multilingual named entity recognition system to Arabic. In: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10), pp. 563–567 (2010)