Natural Language Engineering **23** (6): 877–903. © Cambridge University Press 2017
doi:10.1017/S1351324917000304

LEMAZA: *An Arabic why-question answering system*[*]

AQIL M. AZMI and NOUF A. ALSHENAIFI
Department of Computer Science, King Saud University, Riyadh 11543, Saudi Arabia
e-mail: aqil@ksu.edu.sa, noalshenaifi@ksu.edu.sa

(*Received 1 October 2015; revised 4 July 2017; accepted 4 July 2017; first published online 24 August 2017*)

Abstract

Question answering systems retrieve information from documents in response to queries. Most of the questions are *who*- and *what*-type questions that deal with named entities. A less common and more challenging question to deal with is the *why*-question. In this paper, we introduce LEMAZA (Arabic for *why*), a system for automatically answering *why*-questions for Arabic texts. The system is composed of four main components that make use of the Rhetorical Structure Theory. To evaluate LEMAZA, we prepared a set of *why*-question–answer pairs whose answer can be found in a corpus that we compiled out of Open Source Arabic Corpora. LEMAZA performed best when the stop-words were not removed. The performance measure was 72.7%, 79.2% and 78.7% for recall, precision and *c@1*, respectively.

1 Introduction

With the explosive growth in the amount of information available on the web, question answering (QA) systems are becoming – more than ever – an important research issue. QA is a field of natural language processing (NLP) that automatically provides an answer for a user-posed question in natural language. Unlike traditional information retrieval (IR) systems and search engines, the goal of QA systems is to return an exact answer to the user's query instead of a set of documents.

In the field of QA, there are many types of questions. The most common is the type of factoid questions, i.e., those asked using the words: *when*, *where*, *how much/many*, *who* and *what*. Factoid questions usually ask about a named entity, such as place, person name, organization, etc., and expect a short identifiable answer. Another type regards the questions that use the words *why* or *how to*, e.g. 'Why is it hot in summer?' or 'How to avoid heart attack?'. These questions are more complex and harder to answer. Figure 1 shows a sample text with several types of questions.

[*]We would like to thank W. Al-Sanie for sharing his RST implementation; and the language specialist for helping us with *why*-question–answer pairs. The first author would like to thank Miss Maryam for her assistance in proof-reading the manuscript. Special thanks to all three anonymous reviewers for their constructive comments, which helped in further improvement of the manuscript. This work was supported by a special fund in the Research Center of College of Computer & Information Sciences (CCIS) at King Saud University for which the authors are thankful.

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

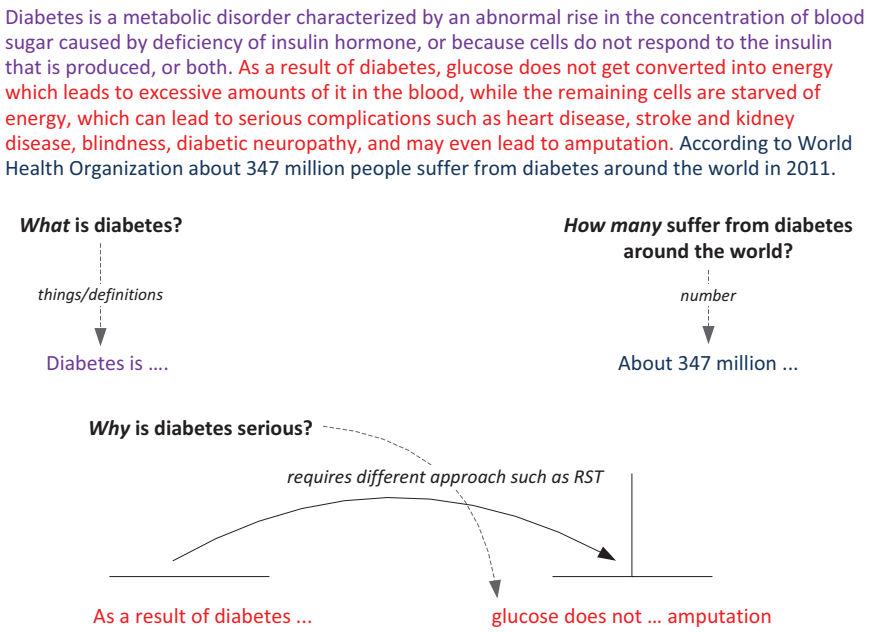


Fig. 1. (Colour online) Example of three different question types: *what*, *how much* and *why*. It helps illustrate the difficulty of answering *why*-questions, and why it is not as straightforward as the more direct type of questions. Sample text is drawn from multiple Arabic sources, translated into English with the aid of Google Translate.

Due to their challenging nature, *why*-questions have received less attention than factoid questions (Verberne *et al.* 2007; Ezzeldin and Shaheen 2012). They require a different approach than factoid questions, as their answers are not named entities, and they tend to be longer and more complex (Ezzeldin and Shaheen 2012). There are few works that have addressed the problem of answering *why*-questions for English, and even less for other languages.

Arabic is a Semitic language and is one of the six official United Nations languages, and the fifth most spoken language in the world with approximately 300 million native speakers (Kanaan *et al.* 2009; Akour *et al.* 2011), along with over 1,500 million worldwide Muslims who use it in their regular daily prayers. Yet, few studies and little effort were directed toward the development of QA system for Arabic language, a modest effort when compared to other languages. Mainly, it is due to the unique aspect specific to the particularities of the Arabic language (Brini *et al.* 2009). And most of the attention in Arabic has been paid to answering factoid questions, in which the answer is a single word or a short phrase.

In this work, we address the task of handling *why*-questions for the Arabic language, a problem that has not been properly addressed in the field of Arabic QA systems. This paper is an extended version of the work published in Azmi and AlShenaifi (2014). We extend our previous work with a more thorough introduction, a look into the characteristic of Arabic language, more detailed description of the system and extra experiments. We propose LEMAZA, a QA system that handles *why*-questions for the Arabic language. The name 'LEMAZA' (لماذا) means *why* in Arabic.

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Our approach to tackle the Arabic *why*-QA problem is composed of the following subtasks: transforming the input question into a query; preprocessing the document collections with the same method used for *why*-question; retrieve candidate passages related to the input question; and extracting the answer. Our approach to extract and generate the answer is based on Rhetorical Structure Theory (RST), which we believe makes for a more feasible way to handle the *why*-questions. The RST is one of the leading theories in computational linguistic and has been successfully applied to a number of NLP linguistic applications. To evaluate LEMAZA, we picked 700 documents in different genres from Open Source Arabic Corpora along with a set of 110 *why*-question–answer pairs. The overall performance measure for LEMAZA is very encouraging with values recall 72.7%, precision 79.2% and correctness at one *c@1* is 78.7%.

The rest of this paper is organized as follows: Section 2 goes over some of the characteristics of Arabic language. Section 3 deals with some of the related works in the area of QA in general and *why*-questions in particular. We do a brief look at discourse segmentation and the RST in Section 4. In Section 5, we present our proposed Arabic *why*-QA system along with an example of how it works. While in Section 6, we discuss the results of evaluating our proposed system. And finally, in Section 7, we conclude the paper with future work.

2 Characteristic of Arabic language

Like all Semitic languages, Arabic is written from right-to-left, and can be classified as classical, modern and colloquial (vernacular). The classical Arabic represents the pure language used by the Arabs, the language the Qur'an was revealed in, while the Modern Standard Arabic is an evolving Arabic with constant borrowing and innovation to meet the modern challenges (Farghaly and Shaalan 2009). The colloquial Arabic was, till recently, confined to daily informal verbal communication and it is only recently that it has started to trickle down into the mainstream written form due to the widespread use of social media (Azmi and Aljafari 2017). A typical Arab communicates using two or more distinct varieties of Arabic in different social contexts. This is a prime example of a linguistic phenomenon known as diglossia (Ferguson 1959). In this work, we will focus on Modern Standard Arabic, the most prevalent form of Arabic found in the written media including the Internet.

Arabic is a phonetic language with twenty-eight basic letters in its alphabet which include three long vowels. There is no capitalization in the Arabic script, however, each letter has more than one shape (up to four), and the shape is position dependent, i.e. initial, middle, final and isolated. The lack of capital letters in Arabic makes it hard to identify proper names, acronyms, abbreviations and further complicates the process of named entity recognition (Farghaly and Shaalan 2009). The orthographic system in Arabic uses small diacritical markings to represent short vowels. The Arabic sound system consists of a total of thirteen different diacritics. The diacritical markings are placed either above or below the letter and are used to indicate the phonetic information associated with each letter, which in turn helps in clarifying the sense and meaning of the word. For example, the undiacritized

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

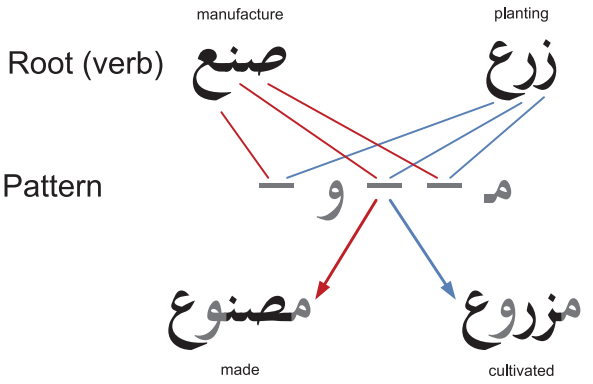


Fig. 2. (Colour online) Example of Arabic language derivations.

word (علم) may mean one of the following (عَلَمْ: *Ealam*) *flag*, (عِلْم: *Eilom*) *knowledge*, (عُلِمْ: *Eulimo*) *became known*, (عَلَّمَ: *Eal~ama*) *taught* or (عَلِّمْ: *Eal~imo*) *teach*.[1] Typically, in Modern Standard Arabic, the diacritical markings are absent from the written text. So, to understand a word, the reader must disambiguate the word based on the context in which it appears. The absence of the diacritical markings produces ambiguity, a task the natives are very good at. As a result, it creates many challenges for the automatic processing of the Arabic language, e.g. word tokenization, word sense disambiguation, question analysis, machine translation, etc. (Azmi and Aljafari 2015).

Arabic is a strongly structured and highly derivational language where morphology plays an important role. The language morphology is a branch of linguistics that defines the internal structure of the words and how they can be constructed. Unlike English morphology which is straightforward, the Arabic language has a systematic but a more complicated morphology. It differs from that of English or other Indo-European languages because it is, to a large extent, based on discontinuous morphemes, primarily consisting of a system of consonant roots which interlock with patterns and vowels to form words, or word stems (Ryding 2005). We can define the root as a relatively invariable discontinuous bound morpheme, represented by typically three (but no more than five) consonants in a certain order, which interlock with a pattern to form a stem that has lexical meaning (Ryding 2005). In Arabic, the two main properties that are used to build words: derivation (see Figure 2) and agglutination. The number of roots in Arabic is finite but huge, the majority of which are triliteral and the longest is quinquiliteral (five-consonant roots). With regard to external morphology, Arabic is an agglutinative language where affixes representing different parts of speech can be conjoined together with a stem or a root to form a word that has a syntactic structure. The agglutination property is not unique to Arabic, and there are other languages that share this phenomenon, e.g. Turkish and Finnish.

[1] For rendering Arabic script into Romanization, we use the Buckwalter transliteration scheme (http://www.qamus.org/transliteration.htm) throughout the paper.

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Arabic is an inflectional language in which affixes representing different parts of speech can be attached to a stem or the root. The words in Arabic take the general form, *prefix(es) + (stem | root) + suffix(es)*. The prefixes can be articles, prepositions or conjunctions, while the suffixes are generally objects or possessive anaphora. For example, the word بمنزلهم (= ب + منزل + هم) which means *in their house*, is composed of a preposition ب, a stem (noun) منزل, and an object pronoun هم. From a statistical point of view, Arabic texts are more sparse because of the inflectional characteristic of the language, which makes each of the NLP tasks more challenging (Benajiba 2007). For instance, in IR, the task is to find the most relevant document(s) to the user's query. If, however, the query keyword appears in a document with additional inflections, then this document will, most likely, be classified as irrelevant. Consider the scenario for the query (عاصمة بريطانيا: *EASmp bryTAnyA*) *Britain's capital*, both words of the query appear in the following text in inflected form (لندن هي العاصمة لبريطانيا الحديثة: *lndn hy AlEAsmp lbryTAnyA AlHdyvp*) *London is the capital for modern Britain*. This text will be classified as irrelevant even though it contains the correct information the user is seeking. We have the same problem with the QA task where the user is interested in an answer to his/her question. The document with the desired answer may not get picked for the same reason as with IR. For more details, see Benajiba (2007).

Another feature of the Arabic language which regards its syntax is that the subject may not be explicitly present, known as pro-drop property (Azmi and Aljafari 2015). Also, Arabic is a free-order language. Though the basic order of Arabic words in a sentence is Verb–Subject–Object, however, it also allows for SVO and VOS orders as well (Farghaly and Shaalan 2009). The sentence, *Mohammad ate the bread* is normally written as أكل محمدٌ الخبزَ (Verb–Subject–Object), but it may also be written as محمدٌ أكل الخبزَ (SVO) and أكل الخبزَ محمدٌ (VOS). Notice that the diacritical markings were used to resolve the Subject and the Object, though in this example we could have done without them using the context only. The absence of the diacritical markings in the sentence, (ضرب الولد الطفل: *Drb Alwld AlTfl*), would prevent to decide who the Subject and Object are, and the sentence could be interpreted as either *the boy hit the child*, or *the child hit the boy*.

3 Related work

The QA system takes a question posted in natural language instead of a set of keywords, understands and analyzes the meaning of the question, and then provides the exact answer from a set of knowledge resources (Hammo *et al.* 2004). The problem of automatically answering questions formulated in natural language has been studied in the field of IR since the mid 1990s (Verberne 2010). But unlike IR, QA system delivers simple and exact answer to a question instead of set of documents from a query (Kanaan *et al.* 2009; Trigui *et al.* 2010). Most of the QA systems are composed of three subtasks: question analysis, passage or document retrieval and answer extraction. Different QA systems may use different implementation for each task (Benajiba, Rosso and Soriano 2007; Ezzeldin and Shaheen 2012).

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

In the field of QA, very little effort was directed toward the development of QA system for the Arabic language, compared to other languages such as English and French (Akour *et al.* 2011). This is mainly attributed to the particularities of the language (see Section 2). The situation is further aggravated by the lack of linguistic resources and NLP tools that is available for Arabic (Abouenour, Bouzoubaa and Rosso 2008; Brini *et al.* 2009). In general, one of the biggest problem with any automatic QA system is the assessment of the answer. This problem spans through all the languages. Recently, there have been some new directions covering this important issue. The first is the automatic machine evaluation of the answers, and second community QA. CLEF, or Conference and Labs of the Evaluation Forum, has many tracks. One of the tracks is QA for Machine Reading Evaluation, which was part of CLEF 2012 and 2013. This track was renamed QA Track starting CLEF 2014. The track is patented on reading test documents and answering multiple choice questions with one correct choice per question (Peñas *et al.* 2012). The answer should be submitted in a specific format so they can be automatically evaluated. The community QA are websites, e.g. Yahoo! Answers, where users turn to looking for an answer or post questions on diverse topics. Community QA system exploits the power of human knowledge to satisfy a broad range of users' information needs, handling factoid as well as complex questions. User interaction in this context is seldom moderated, and as thus, there is little restriction (if any), on who can post and who can answer a question. Nevertheless, this is a valuable resource. For some of the recent work on answer selection in CQA, see Nakov *et al.* (2015, 2016, 2017).

Next, we briefly go over some of the QA systems in general with a special focus on *why*-questions, followed by a look into Arabic QA systems with a detailed description of existing approaches on Arabic *why*-QA.

3.1 General look into non-Arabic QA systems

As we noted, most of the research in the area of QA has focused on answering factive questions such as factoids and definitions in which named entity recognition plays an important role in identifying correct answers (Verberne *et al.* 2011). However, we see very little attempt to build QA system that answers *why*-questions. This should not come as a surprise, since *why*-questions tend to be complex questions that require a different technique and their answers are passages that give an explanation (possibly implicit) which cannot be stated in a single phrase (Verberne *et al.* 2010). No doubt, most of the research in the field of IR including QA has been confined to the English language (Kanaan *et al.* 2009). Bosma (2005) showed how the existing answer of a QA system can be improved by exploiting RST-based summarization technique. One of the critical steps in the design of QA systems is the passage retrieval (PR) and ranking module. The current state of the art work in QA is to re-rank the candidate answers, a scheme which improves the overall accuracy of the QA system. In a series of papers, the authors went on using different schemes to rank the candidate answers using supervised classifier (Support Vector Machine) (Severyn and Moschitti 2012), convolutional deep neural networks (Severyn and Moschitti 2015) and tree kernel (Tymoshenko and Moschitti 2015). In the latter paper, the authors propose using

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

syntactic structures and semantic information from the classifier and the knowledge from Linked Open Data for re-ranking the answer passages.

There are few works that explicitly address *why*-QA. Verberne *et al.* (2007) explained a method for *why*-QA that is based on discourse structure and relations in a pre-annotated document collection (the RST Treebank). Answers to *why*-questions can be extracted by matching the question topic to a span in RST-tree and selecting the most relevant answer according to the RST relation that holds between a question topic and its answer spans. Later, the authors described an approach for ranking answers to *why*-questions by evaluating a number of machine learning techniques in their performance that are described by a set of thirty-six linguistically motivated features (Verberne *et al.* 2011). In Verberne *et al.* (2010), the authors proposed a *why*QA system that preprocessed the input *why*-question to produce a query, then they used an off-the-shelf PR system to retrieve texts that were relevant to the input query, after which they applied a re-ranking module to return a ranked list of candidate answers. Higashinaka and Isozaki (2008) proposed a corpus-based approach for answering *why*-questions. The scheme is to automatically collect causal expressions from a tagged corpora with semantic relations, which will be used to train a ranker of the candidate answers. The authors presented NAZEQA, a Japanese *why*-QA system based on their approach. Motivated by the observation that the answer to *why*-questions often follow a specific pattern, the authors explored utilizing sentiment analysis and semantic word classes to improve Japanese *why*-QA on a large-scale web corpus (Oh *et al.* 2012). The authors in (Oh *et al.* 2013) explored the utility of intra- and inter-sentential causal relations for ranking the candidate answers to *why*-questions. This helped improve their precision by 4.4% over their earlier work (Oh *et al.* 2012).

3.2 Arabic QA systems

For the Arabic language, most of the work in the area of QA has focused on answering factoid questions. For a survey on the existing Arabic QA systems and tools, the reader is advised to consult (Shaheen and Ezzeldin 2014). We will go over some of the systems in more depth. Hammo *et al.* (2002) proposed QARAB, a QA system that provides short answers to questions written in Arabic language. The knowledge source of the proposed system is a collection of Arabic newspaper. QARAB utilizes well-known techniques from IR to identify the candidate documents by treating the question as a query and then applies keyword-matching strategy. It also employed NLP techniques to syntactically parse the question and analyze the top-ranked textual documents retrieved by the IR system. QARAB excluded queries which start with *how* and *why*-questions because they require a more complex processing. It achieved a precision and a recall of 97.3%, where the results was manually evaluated. Four native Arabic speakers presented 113 questions to the system, who in turn also assessed the correctness of the system-produced answers. An issue that raises some doubt about QARAB is its reported performance measure. It is high by any standard, and we did not come across any QA system in any language with such a performance. Brini *et al.* (2009) described Question-Answering

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

System for Arabic Language which handles factoid and definition questions using NooJ platform (Silberztein 2005). Question-Answering System for Arabic Language takes a question expressed in natural language and produces the most efficient answer. It uses IR and NLP techniques to process Arabic factoid questions along with a collection of Arabic documents. DefArabicQA is an Arabic definitional QA system by Trigui *et al.* (2010), one of the first to focus on such subject for the Arabic language. Definitional questions take the form 'what is X?', and these are generally geared toward information about an organization or an entity. Usually the system identifies the answer, which is a definition about the organization/entity from Web resources. DefArabicQA was implemented based on pattern approach and employs a little linguistic analysis. Rosso, Benajiba and Lyhyaoui (2006) described a generic architecture for an incomplete Arabic QA system. The authors implemented some of the components and presented the results of testing them. They implemented named entity recognition system and PR module, both constitute an important part of a QA system. The PR was adapted from an existing language-independent PR system. To finish their implementation, they need to build two more modules, question analysis and the answer extraction modules.

Salem *et al.* (2010) devised an RST-based system to answer *why* and *how to* questions expressed in Arabic natural language. The authors tested their system on Arabic raw texts drawn from Arabic websites, where the text was automatically annotated. They reported a recall of fifty-five% for ninety-eight question–answer pairs covering *why* and *how* questions. Akour *et al.* (2011) introduced a rule-based QA system QArabPro for reading Arabic comprehension tests in Arabic language. Here, the authors considered all types of questions and did not shy away from *how* and *why*. The dataset consisted of seventy-five reading comprehension tests collected out of Wikipedia (Ar), and 335 questions which included forty-five *why*-questions. The system achieved an overall accuracy of 84.18% on the full test set, however, the accuracy for *why*-questions was 62.22%.

4 Discourse segmentation and RST

Discourse analysis investigates how texts are organized and tries to grasp their underlying structure. The structure which spans a few sentences is often called a discourse segment. Determining the segment boundaries is theory dependent as each theory defines its own segmentation guidelines and unit size. Some of the main discourse theories are RST (Mann and Thompson 1988), Segmented Discourse Representation Theory (Asher and Lascarides 2003) and Discourse Lexicalized Tree-Adjoining Grammar (Webber 2004). The discourse in Arabic tends to use sentences which are long and complex, and it is not uncommon to find an entire paragraph without any punctuation. Adapting the discourse theories for Arabic language is not a trivial task. Some of the adaptations are Al-Sanie (2005) for the RST theory, and Keskes *et al.* (2014) for the Segmented Discourse Representation Theory. Though the latter segmentation scheme is more fine grained than the RST, we however, will be using RST as it has been around for a while and was applied in different NLP applications, e.g. text generation (Scott and de Souza 1990), summarization

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

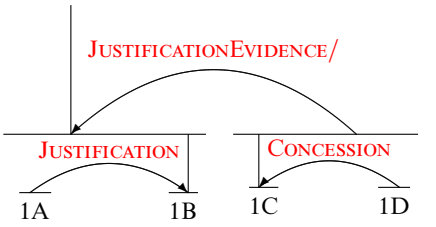
(Marcu 1998; Azmi and Al-Thanyyan 2012) as well as in QA (Salem *et al.* 2010; Verberne 2010). As RST is an essential part of our work, we feel it is necessary to briefly go over it. RST was originally developed in late 1980s, and provides a framework for describing texts and rhetorical relations among parts of a text (text span); it identifies hierarchic structure in text. There are two underlying principles to RST: (1) coherent texts consist of minimal units that are linked (recursively) to each other through rhetorical relations; and (2) coherent texts do not show gaps, i.e. there should be some relation attributable to different parts of the text (Taboada and Stede 2009).

In RST, the first step in analyzing a text is to divide it into many spans (or discourse units) in an arbitrary size and this will be the smallest units of discourse. The process of parsing the text and constructing the rhetorical structure is known as the rhetorical analysis. During this process, we determine the spans (text units) that participate in constructing the rhetorical schema and the rhetorical relations that hold these units. One method used to extract text spans and mark the structure of a discourse is cue phrase-based approach. Cue phrases are words and phrases that are considered as unit connectors. They are used to connect adjacent sentences, and are important for the reader to understand the text. A span can be classified as either *nucleus* or *satellite*. Nucleus is the span that is most essential to text, while satellite contains additional information about the nucleus. The satellite in turn, may be another nucleus (multinucleus) or just a simple satellite. Satellites may be removed without affecting the meaning of the text. However, nucleus cannot be removed because it holds the most important information for understanding the text. The next big step is to identify rhetorical relations that hold between text spans. There are two types of rhetorical relations: asymmetric relations and symmetric relations. In the former, one span is considered nucleus and the other related span is the satellite, while for the latter all spans are of equal status, i.e. they both have the same importance. An example of a symmetric relation is the relation CONTRAST, here we cannot consider a span more important than the other (Bateman and Delin 2006).

The rhetorical relations are defined as constraints or conditions which holds between two non-overlapping text spans. Each relation consists of four fields (Mann and Thompson 1988): (a) set of conditions on the nucleus; (b) set of conditions on the satellite; (c) set of conditions on the combination of nucleus and satellite and (d) specification of the effect. Mann and Thompson (1988) defined a set of twenty-three rhetorical relations for English. By identifying relations between spans of text, a full rhetorical structure representation known as RS-tree is created, where leaf node represents text span and the intermediate node is the rhetorical relation that holds between the two spans. It is possible to have multiple analysis and hence several RS-trees for a single text (Seif *et al.* 2005). The set of rhetorical relations are not universal among languages. Following an investigation of the relations in Arabic corpus and in literature, Al-Sanie (2005) concluded that not all rhetorical relations have an equivalent in Arabic. For example, the relations CONCESSION and CONTRAST are not known in the Arabic rhetorics, and the closest Arabic relation is RECALLING. In the end, there are a total of eleven rhetorical relations in Arabic compared

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Table 1. *A single sample text and two different RS-trees*

[No matter how much one wants to stay a non-smoker,]^{1A} [the truth is that the pressure to smoke in junior high is greater than it will be any other time of one's life.]^{1B} [We know that 3,000 teens start smoking each day,]^{1C} [although it is a fact that 90% of them once thought that smoking was something that they'd never do.]^{1D} (text$_{0786e}$)	

to twenty-three rhetorical relations in English. Following is a list of the Arabic rhetorical relations as reported by Al-Sanie (2005) along with the relations name in Arabic: (CONDITION: شرط), (INTERPRETATION: تفسير), (BASE: قاعدة), (JUSTIFICATION: تعليل), (EXPLANATION: تفصيل), (CONFIRMATION: توكيد), (RESULT: نتيجة), (EXAMPLE: تمثيل), (RECALLING: استدراك), (JOINT: عطف) and (SEQUENCE: ترتيب).

We can build the RS-tree as soon the text units are identified, and the rhetorical relations holding between the units are recognized. Let $U = u_1, u_2, \ldots$ be a sequence of textual units, and $\mathscr{R}$ a set of rhetorical relations that hold among these units and among contiguous textual spans over U. Marcu (2000) provided a complete formalization of the RS-tree using the predicates: *rhet_rel*(R, u_i, u_j) which is true if the rhetorical relation R is consistent with the relation that spans between the textual unit u_i (mostly satellite), and the nucleus unit u_j. Multinuclear relations are represented using the predicate, *rhet_rel_ext*(R, s_s, s_e, n_s, n_e) where R spans textual units s_s–s_e and n_s–n_e, where one or both textual spans are longer than one unit. Marcu provided a proof-theoretic account that generates all valid RS-trees. He goes on presenting axioms along with an algorithm that uses the given axioms to derive all the valid RS-trees. Table 1 is a sample text and its corresponding RS-trees taken from Marcu (1997). Applying the definitions in Mann and Thompson (1988), we get five rhetorical relations for this sample text: *rhet_rel*(JUSTIFICATION, $1A, 1B$); *rhet_rel*(JUSTIFICATION, $1D, 1B$); *rhet_rel*(RESTATEMENT, $1D, 1A$); *rhet_rel*(CONCESSION, $1D, 1C$) and *rhet_rel*(EVIDENCE, $1C, 1B$). These relations hold because the understanding of both 1A and 1C increases the reader's willingness to accept the writer's statement 1B; the understanding of 1C will increase the reader's belief of 1B etc. From the five rhetorical relations and the axioms, the author presented four different RS-trees can be constructed, two of which are shown in Table 1. Note that one of the RS-trees has JUSTIFICATION as the rhetorical relation linking textual span 1C–1D and 1A–1B, and the other RS-tree has EVIDENCE along the same textual span.

5 Our proposed Arabic why-QA system

We have seen in Section 3.1 that RST was successfully used for answering *why*-questions in English, e.g. Verberne (2010). There are two issues at hand: (1) the

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

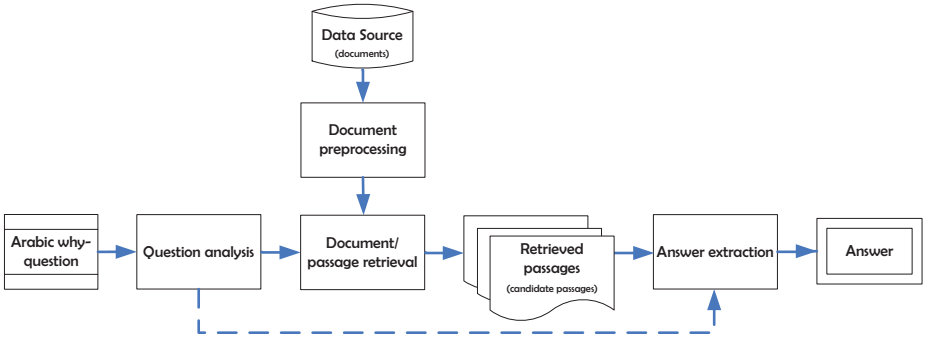


Fig. 3. (Colour online) General architecture of our Arabic *why*-question answering system.

drastic difference in the number of rhetorical relations between English and Arabic; and (2) occasionally, the same passage may have different rhetorical structure when conveyed in different languages (Iruskieta, da Cunha and Taboada 2014). These two issues are a cause to worry about the effectiveness of RST in answering Arabic *why*-questions. In this work, we hope to show that the success of RST in answering *why*-questions in English does transcend the language barrier to Arabic with equal success. QA systems deal with different types of questions. Most of the efforts were focused on factoid questions, e.g. *when* and *where*. Questions using the words *why* and *how*, are hard to answer and require deep processing. Our goal is to develop *why*-QA system for the Arabic language. The input to our system is a *why*-question expressed in Arabic natural language and the output is a paragraph that contains the most efficient and appropriate answer. In Arabic language, questions starting with the word (لماذا: *lmA*A*) *why*, and (ما سبب: *mA sbb*) lit. *what is the reason*, both designate a *why*-question, and our system handles them both. The question takes on a slightly different form when using one or the other expression. For example, (لماذا أرتفع سعر الذهب: *lmA*A ArtfE sEr Al*hb*) *Why did the gold price go up?*, and (ما سبب أرتفاع سعر الذهب: *mA sbb ArtfAE sEr Al*hb*) *What is the reason behind high gold price?*, both forms ask the same question.

Figure 3 shows the general architecture of our Arabic *why*-QA system. The system is composed of four main components: question analysis, document preprocessing, document/PR and answer extraction. In the first component, NLP techniques are used to parse and process the input question in order to get some useful information and formulate the query. Documents will also be processed in a similar way by the second component. The third component will retrieve a list of candidate passages that may possibly contain the answers. Finally, the last component, answer extraction, returns the answer to the user. For further details on the individual components, see Sections 5.1–5.4. Our approach of tackling *why*-QA problem is oriented for open-domain non-structured documents under the assumption that the answer exists somewhere in our corpus of Arabic textual documents.

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

5.1 Question analysis

The main task of this component is to preprocess the question in order to extract the question keywords and obtain some useful information which represents the user's need (Rosso *et al.* 2006). In general, this component relies on NLP tools that perform linguistic analysis on the question and then generates a query. It treats the *why*-question as a 'bag of words' in order to retrieve a list of ranked documents that may possibly contain the answer. So far, there is no question analysis procedure that has been created for *why* QA specifically, neither for Arabic nor English (Verberne 2010). We, however, will develop an approach for *why*-question analysis which is based on existing modules of factoid question analysis. It will also allow us to figure out whether the question analysis procedure used in factoid questions is suitable for non-factoid or *why*-questions. Our approach is to use shallow language understanding using Arabic NLP tools to analyze the *why*-question. We do not attempt to semantically understand the content of the *why*-question at a deep level.

Algorithm 1: The proposed algorithm for the question analysis. This constitute the first component (out of four) that makes up our Arabic *why*-QA system.

Input: *Why*-question *S*

1 **begin**
2 Tokenize *S* to extract words or individual terms.
3 Normalize *S* by transforming variants of a character into single form.
4 Stop-word removal (optional).
5 Apply stemming algorithm to obtain the roots.
6 Formulate and generate the query.
7 Expand the query by including synonyms and words that share the same root.

The question analysis is a six step process (Algorithm 1). In the text normalization step, we rewrite the question *S* by unifying all variants of a letter into a single form, e.g. (أ، إ، آ → ا, ؤ → و, ة → ه etc), we also strip the text *S* of all the diacritical markings. The stop-word removal is optional, and as we will see later, it is better to keep them (see Figure 6). El-Khair (2006) is a good source for a list of Arabic stop-words. What we have now is a set of words that needs to be stemmed to become keywords. In step 6, we generate a query using the keywords and these are passed to the PR component (see Figure 3). In query expansion, the last step in Algorithm 1, we extend the list of extracted keywords by adding new words that connect semantically to those in the query. This is achieved by: (1) using Arabic dictionary of synonyms; and (2) adding other forms of the words that share the same root. The result of the query expansion is again passed to PR component to retrieve a ranked list of passages that match the query's words. This step helps to improve the retrieval results, as some studies have shown that query expansion increases the recall of QA task (Hammo *et al.* 2004; Ezzeldin and Shaheen 2012). For the dictionary, we use Arabic WordNet (http://globalwordnet.org/arabic-wordnet/), a multi-lingual concept dictionary that provides mapping between word senses in Arabic and those

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Table 2. *Example of how a question is processed by Algorithm 1. We do not show the whole set of expanded keywords*

Token	Normalize	Stop-word	stem	Expanded keywords
(لماذا: *lmA*A*)	لماذا	No	اذا	
(يحذر: *yH*r*)	يحذر	No	حذر	(يحذر: *yH*r*), (تحذير: *tH*yr*)
(من: *mn*)	من	Yes		
(كثرة: *kvrp*)	كثره	No	كثر	(كثرة: *kvrp*), (الكثير: *Alkvyr*)
(استخدام: *AstxdAm*)	استخدام	No	خدم	(خدمة: *xdmp*), (استغلال: *AstglAl*), (استخدام: *AstxdAm*), (استعمال: *AstEmAl*), (تسخير: *tsxyr*), (وظيفة: *wZyfp*), (توظيف: *twZyf*)
(مستحضرات: *mstHDrAt*)	مستحضرات	No	حضر	(محضر: *mHDr*), (مستحضر: *mstHDr*), (مستحضرات: *mstHDrAt*)
(التجميل: *Altjmyl*)	التجميل	No	جمل	(جميل: *jmyl*), (تجمل: *tjml*), (التجميل: *Altjmyl*)

in the Princeton WordNet (English), which also supports searching and browsing of Arabic and English terms.

Due to the morphological complexity of the Arabic language, some kind of morphological processing would help recover the units of meaning or their proxies, such as stems (or roots). Stemming is the process of conflating various forms of the same word to its stem or root form. It has been shown that stemming improves IR in Arabic (Semmar *et al.* 2006). In our case, stemming helps resolve the problem of possible mismatch between the form of a word in *why*-question and its form in candidate documents. The two broad approaches to stemming are root-based approach (Khoja and Roger 1999), and stem-based (or light stemming) approach (Larkey, Ballesteros and Connell 2002). There is a consensus that light stemming is easier and more accurate than its counterpart, root-based stemming, as it is restricted to stripping off a predetermined set of Arabic affixes. However, such a notion is not true. In reality, light stemmer struggles when the initial part of the Arabic word resembles a prefix, e.g. (والي: *wAly*) *governer* and (وباء: *wbA'*) *epidemic* (Al-Kabi *et al.* 2015). For stemming, we use the root-based stemmer (Khoja and Roger 1999). What gave impetus to our decision are the different studies which confirm that indexing Arabic text using word's root form significantly increases retrieval effectiveness (Habash and Rambow 2005; Salem *et al.* 2010), and that some of the Arabic QA systems make use of root-based stemmer, e.g. QARAB (Hammo *et al.* 2002) and QArabPro (Akour *et al.* 2011). We did a side experiment using a different stemmer, Morphological Analysis and Disambiguation for Arabic in particular, but failed to observe any impact on the overall performance (see Section 6).

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

To illustrate Algorithm 1, we show how a question is processed. Table 2 applies the individual steps of Algorithm 1 to the question, (لماذا يحذر من كثرة استخدام مستحضرات التجميل؟ :*lmA*A yH*r mn kvrp AstxdAm mstHDrAt Altjmyl?*) *Why do they warn us from using too much cosmetics?*, producing a set of expanded keywords.

5.2 Document preprocessing

Prior to using PR stage, we need to process the textual documents in the data source (Figure 3). The idea is to apply the same processing the *why*-question query goes through. This guarantees a common ground. Document preprocessing includes the following steps: tokenization, normalization, stop-word removal and stemming. See Section 5.1 for details regarding individual steps.

5.3 Document/passage retrieval

PR is a core component in any QA system. The success of a QA system depends heavily on the quality and effectiveness of PR component; and if retrieving process fails in finding the relevant passages for the *why*-question, then it is not possible to proceed further (Brini *et al.* 2009). PR uses the query generated by an earlier component, question analysis (Section 5.1), to extract and retrieve a ranked list of candidate passages. This means retrieving passages with the highest probability of containing expected answer to user's query (Rosso *et al.* 2006; Kanaan *et al.* 2009).

Researchers have investigated different techniques for the PR. In Rosso *et al.* (2006) and Benajiba *et al.* (2007), the authors adapted the language-independent JAVA Information Retrieval System PR system for Arabic which relies on n-gram model to index the documents. JAVA Information Retrieval System uses distance density model to rate the retrieved passages (Ezzeldin and Shaheen 2012). Akour *et al.* (2011) constructed an IR system based on Salton *et al.* (1975) statistical Vector Space Model, while Zhao, Xu *et al.* (2006) proposed a retrieval module composed of documents and passages retrieval. The documents retrieval module was based on System for the Mechanical Analysis and Retrieval of Text, a retrieval system developed under G. Salton at Cornell University; while the passages were retrieved based on system similarity model (ISSM) between questions and sentences in relevant documents. On the other hand, Verberne (2010) used two different engines, a retrieval engine Lemur (http://www.lemurproject.org/) and a search engine Wumpus (http://www.wumpus-search.org/), both are considered off-the-shelf engines.

For our system we used Lemur, an off-the-shelf retrieval module for the PR component. The Lemur Project was started in 2000 by the Center for Intelligent Information Retrieval at the University of Massachusetts, and the Language Technologies Institute at Carnegie Mellon University. Lemur is an IR toolkit developed to assist research in language modeling, text mining and IR. There are a few reasons why we decided to use Lemur as an IR system. These can be summarized as Lemur supports cross-lingual retrieval; a built-in support for text documents written in

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Arabic language; provides multiple methods for indexing small- and large-scale collections; supports PR rather than retrieving the whole document; works with un-/semi-/structured document retrieval; and there are APIs available for C++, Java and C#. The candidate passages retrieved by this component are passed to the final stage (answer extraction) for further processing and extracting an exact answer. Using a structure-based PR module is not enough for a robust retrieval task. Therefore, as we previously mentioned, we use query expansion module in the question analysis component (Section 5.1) in order to overcome the situation where there is a mismatch between question keywords and the words in the document. In other words, to avoid the case in which the PR module eliminates relevant passages containing other forms or other interpretations of the question keywords (Abouenour, Bouzouba and Rosso 2013).

5.4 Answer extraction

Once the relevant passages are retrieved, they are further processed. The main goal of this component is to return the appropriate answer extracted from the retrieved passages by the previous component, PR. The input to answer extraction component is the 'bag of words' of *why*-question and a list of ranked relevant passages. Certainly, if the retrieved passages are not relevant to question's keywords and do not contain the answer, this component will fail to return the exact answer. For the answer extraction module, Verberne (2010) proved that RST-based method improves the recall measure of the answer of a *why*-question written in English. In this work, we will follow the latter's lead and use RST-based method for answering Arabic *why*-questions. The Arabic RST engine we use is the one devised in Al-Sanie (2005).

In Section 4, we briefly went over RST. A set of twenty-three rhetorical relations have been defined for English (Mann and Thompson 1988). Given our focus on answering Arabic *why*-questions, we need to extract a set of Arabic rhetorical relations that helps achieve our goal. Out of eleven rhetorical relations in Arabic (Al-Sanie 2005), only five relations are related to reasons and causality sentences in the Arabic language, making them relevant to our *why*QA system. These are INTERPRETATION, JUSTIFICATION, RESULT, EXPLANATION and BASE.

Before constructing rhetorical relations, we first need to determine text spans/units in the candidate passages retrieved by PR component and then find the relations that hold between these units. We use cue phrases for rhetorically parsing candidate passages in order to indicate the boundaries between elementary units of texts and the rhetorical relations that hold between them (Seif *et al.* 2005; Salem *et al.* 2010). Cue phrases are words and phrases that are considered as unit connectors. Different cue phrases can indicate different relations. For example, the relation RESULT can be hypothesized on the basis of the occurrence of the cue phrases (لذا، نتيجة لذلك، نتيجة لهذا، نستنتج). Similarly, the relation JUSTIFICATION related on the cue phrases (تعليل، بسبب، لأن). So we use rhetorical parsing algorithm whose input is a free unrestricted Arabic text, which is a candidate passage, and produces rhetorical structure as an output. We need to ensure that the *why*-question keywords

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

correspond to a text unit and the answer corresponds to another text unit; and an RST relation holding these two text units. The text unit that needs explanation is the nucleus of the relation, and the text unit giving the explanation is the satellite. The only exception to this rule is the *result* relation, which is the other way around. The RST-based procedure to extract answer for the *why*-question is listed as Algorithm 2.

Algorithm 2: The RST-based algorithm to extract answer for the *why*-question.

Input: Arabic text T, question query Q, and question expanded query EQ
Result: Answer or empty (failed to find an answer)

```
1 begin
2     Identify the set of all cue phrases in T.
3     Identify the elementary units in T based on the cue phrases (determined in previous
      step) and text markers.
4     Hypothesize rhetorical relations between the elementary units.
5     Build all RS-trees for text T.
6     while we have more text units in RS-tree representing Q and EQ keywords do
7         Is the found unit the nucleus of rhetoric relations {INTERPRETATION,
          JUSTIFICATION, EXPLANATION, BASE} (or, in case of RESULT relation, the satellite)?
          If NO, continue with the next iteration.
8         return related satellite (or nucleus in case of RESULT relation) of the found unit
          as an answer.
9     return no answer.
```

5.5 Main algorithm

Our proposed algorithm to handle the Arabic *why*-questions is listed as Algorithm 3. Since we have five rhetorical relations that are relevant to our *why* QA system, our implementation focuses on retrieving answer based on the priority of RST relations. We designate three priorities, the highest priority is given to the relation JUSTIFICATION, the least priority is for the relation BASE, and the remaining three have the same priority. So for example, if we have two candidate answers, one corresponds to relation JUSTIFICATION and the other to the relation INTERPRETATION; in this case, we retrieve the answer to JUSTIFICATION as it has higher priority. In case the candidate answers all have the same RST relation priority, then we randomly select one of the candidate answers.

The next example helps illustrate our algorithm. Consider the sample Arabic text (Table 3), and the corresponding RST-tree (Figure 4). Suppose we issue the query, لماذا يحذر من خطورة استخدام مستحضرات التجميل؟ *Why do they warn us from using cosmetics?* We have two candidate answers (see Table 4). The first candidate answer corresponds to RST relation JUSTIFICATION, while the other candidate answer corresponds to the RST relation RESULT. Our system will pick the first candidate answer since it corresponds to an RST relation with higher priority.

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Algorithm 3: The main algorithm to handle Arabic *why*-questions.

Input: Arabic *why*-question S
Result: Answer or empty (failed to find an answer)

begin
  Tokenize S.
  Normalize text S.
  Extract question keywords by eliminating stop-words.
  Extract the root of keywords in S.
  Convert S into query Q.
  Expand keywords in S.
  Convert S into expanded query EQ.
  Index the queries Q and EQ.
  Find index weight of each indexed term (keywords) in Q and EQ.
  T = retrieved passages.
  return the result returned by Algorithm 2.

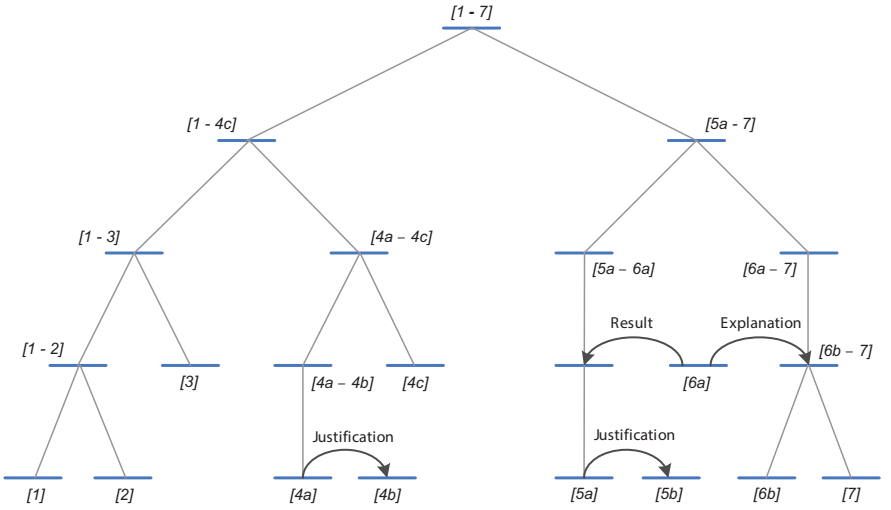


Fig. 4. (Colour online) Corresponding RST-tree to the sample text featured in Table 3.

6 Results and discussion

We implemented our Arabic *why* QA system using Java. To test our system, we compiled a corpus of 700 articles of different genres, all extracted from Open Source Arabic Corpora (Saad and Ashour 2010). In compiling the corpus, we did not tamper with the contents even if there were grammatical mistakes. The system was evaluated by an Arabic native speaker (a specialist who is not one of the authors). To guarantee an answer, we asked the specialist to go over the corpus before formulating the set of *why*-questions. Posing questions blindly has a good chance that the answer may not be there. The person posing the question has no clue on the exact content nor the detail found in the document. We collected 110 *why*-question

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Table 3. *A sample Arabic text titled, 'Toxic metals are a main component in commercial and counterfeit cosmetics'*

[يظن غالبية مستخدمي مستحضرات التجميل أنها مفيدة للبشرة وحيويتها ونضارتها وشبابها] [1] [ولا يعرف أن هذه المستحضرات قد تكون وراء ظهور أمراض عديدة مثل الفشل الكلوي والكبدي والتهابات في العيون وتساقط في أهداب العيون والتهابات في الشفاه والفم واللثة] [2] [وقد تكون مستحضرات التجميل وراء حساسية الجلد وتشققات في البشرة وأحمرارها] [3] [هذه الخطورة والتفاقم الصحي المتسبب من مستحضرات التجميل،] [4a] [لاحتوائها على الرصاص والزئبق والزرنيخ والكادميوم وكلها معادن سامة وخطيرة عرف تأثيرها على أعضاء الجسم المختلفة وقد تكون مكونات المستحضرات التجميلية لها دور في حدوث الحساسية والحكة واحمرار الجلد وقد يكون بتلوث هذه المستحضرات التجميلية بالبكتيريا والجراثيم لها دور في التهابات العيون والجفون والتهابات في الجهاز الهضمي وقد تكون هذه الجراثيم وراء ندبات والتهابات الجلد المختلفة،] [4b] [هذه الأمراض المختلفة تجعل الكثير يتنقلون بين عيادات الأطباء والمراكز الطبية وهم لا يعلمون أن هذه المستحضرات التجميلية هي وراء الحساسية والحكة والتحرش واحمرار والطفح الجلدي أو تشققاته أو تكون وراء الفشل الكلوي والكبدي والذي يكتشف بعد فترة طويلة من استعمال هذه المستحضرات التجميلية والملوثة بالملوثات المختلفة والتي أظهرت نتائج التحليل بالأجهزة العلمية الحديثة والدقيقة أو أظهرتها المزارع البكتيرية والتحاليل الجرثومية.] [4c] [وقد تحفظ مستحضرات التجميل في أماكن رطبة وغير باردة فتكون بيئة صالحة لنمو البكتيريا والجراثيم وقد تحفظ في الشمس والأماكن الحارة فيتغير تركيبها وتحصل تفاعلات كيماوية لهذه المستحضرات فتصبح مواد خطيرة،] [5a] [مسببة لالتهابات جلدية يصعب علاجها بسهولة.] [5b] [لذا ننصح ونوصي بأن تباع وتوزع مستحضرات التجميل في أماكن تحت إشراف متخصصين وفي مراكز معروفة مثلها مثل الأدوية والمستحضرات الصيدلانية ويراقب منتجو ومصنعو مستحضرات التجميل ويحاسب من يخل أو يخطئ بهذه النهنة المهمة والتي يتداولها مختلف الأعمار من رجال ونساء وأطفال ويصرف المجتمع ملايين الدولارات سنوياً على مستحضرات التجميل لذلك وضع رقابة على تصنيعها وبيعها وتداولها أمر مهم يجدر بالوزارات المعنية تطبيقه والأهتمام به حفظاً لصحة المستهلك وحفظا لأمواله فالصيدليات يعتني بها صيادلة والبصريات والنظارات الطبية يعتني بها مختصون ومستحضرات التجميل يجب أن يعتني بها مختصون،] [6a] [وخاصة أنها توضع في أماكن مهمة وحساسة مثل العيون والجلد والخدود والشفاه ولمختلف الأعمار من الرضع حتى المسن وخاصة مستحضرات التجميل كما سبق وأن ذكرنا العديد منها مثل الكحل وبودرة الشفاه وبودرة الوجه ملوثة بالرصاص والزرنيخ وملوثة بالبكتيريا والجراثيم المرضة.] [6b] [لذلك يجدر أن يبيعها ويوزعها ويشرف عليها مختصون ومراكز طبية مختصة.] [7]

Passages are bracketed and numbered for referencing. The corresponding RST tree is in Figure 4. Text source: `http://www.alriyadh.com/463796`

and answer pairs.[2] The collected questions were then passed to LEMAZA in order to get its answer. To judge the answer, we compared both of them; the one returned by the system and the one formulated by the specialist during manual analysis. Only if both agree, will we consider it as a correct answer. Figure 5 is a screenshot of our *why* QA application.

Precision P and recall R are standard measures that are used to evaluate the IR systems performance. Gaizauskas and Humphreys (2000) redefined P and R for QA systems. To evaluate our system's performance, we will use these two redefined measures in addition to a third measure. Though recall is an important indicator for measuring the success of a QA system because most of the users are interested

[2] We are sharing the full dataset with the interested researchers. Please contact the corresponding author.

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Table 4. *Two candidate answers for the Arabic query 'Why do they warn us from using cosmetics?', corresponding to different RST relations. Our system picks candidate answer 1 since its RST relation has a higher priority*

Candidate answer 1	Candidate answer 2
[لاحتوائها على الرصاص والزئبق والزرنيخ والكادميوم وكلها معادن سامة وخطيرة عرف تأثيرها على أعضاء الجسم المختلفة وقد تكون مكونات المستحضرات التجميلية لها دور في حدوث الحساسية والحكة واحمرار الجلد وقد يكون بتلوث هذه المستحضرات التجميلية بالبكتيريا ...]	[لذا ننصح ونوصي بأن تباع وتوزع مستحضرات التجميل في أماكن تحت إشراف متخصصين وفي مراكز معروفة مثلها مثل الأدوية والمستحضرات الصيدلانية ويراقب منتجو ومصنعو مستحضرات التجميل ويحاسب من ...]
Because they contain lead, mercury, arsenic and cadmium, all toxic metals and dangerous it has a side effect on various body organs; and it is possible that the components of the cosmetic products may lead to allergies, itching and redness of the skin; and the contamination of these cosmetic products with bacteria …	So we advise and recommend that cosmetics must sell and distribute in special places under the supervision of specialists in centers, just like drugs and pharmaceuticals; and the producers and manufacturers of cosmetics should be watched, and punish those …
RST relation: JUSTIFICATION	RST relation: RESULT

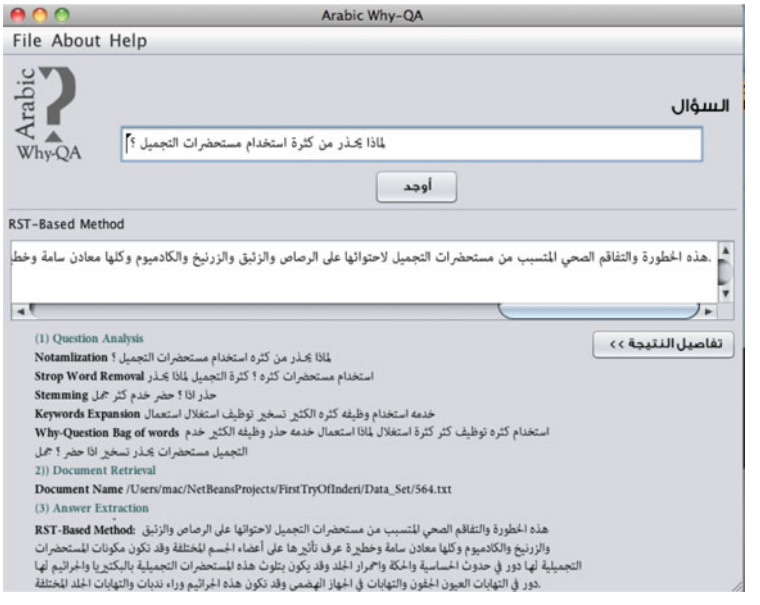


Fig. 5. (Colour online) Screenshot of LEMAZA, our Arabic *why* QA system.

in a correct answer rather than just an answer. However, neither of the redefined measures capture the whole picture. Usually, in a real-life situation, we would rather have a no answer than an incorrect one. Based on a simple yet rigorous analysis, Peñas and Rodrigo (2011) proposed *c@1*, a measure that discriminates between

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

a question that is unanswered and one which is wrongly answered. In a way, it is to encourage the development of a system that can check the correctness of its response (Peñas *et al.* 2012). This measure was adopted in QA for Machine Reading Evaluation which was part of CLEF 2012 and 2013. All three measures are expressed as numbers ranging between 0 and 1 inclusive. Defining the measures formally, Equations (1) and (2) are given in Gaizauskas and Humphreys (2000), and Equation (3) in Peñas and Rodrigo (2011),

$$P = \frac{\#\text{ correct answers}}{\#\text{ questions answered}} \tag{1}$$

$$R = \frac{\#\text{ correct answers}}{\text{total }\#\text{ questions}} \tag{2}$$

$$c@1 = \frac{\#\text{ correct answers}}{\text{total }\#\text{ questions}}\left(1 + \frac{\#\text{ unanswered questions}}{\text{total }\#\text{ questions}}\right) \tag{3}$$

$$= R \cdot (2 - R/P) \tag{4}$$

where Equation (4) is derived by substituting the definitions of P and R in Equation (3). A variant of the first measure is precision at k, denoted $P@k$ (Manning, Raghavan and Schutze 2008). It refers to the proportion of relevant documents in the first k positions. So, $P@1$ refers to precision of the top answer, a measure of how many questions have a correct top-answer candidate. This measure was used to report the performance of answering *why*-questions in Japanese (Oh *et al.* 2013). Since Lemaza only reports the top answer, hence, we will be using $P@1$, recall R and $c@1$ to evaluate our system. Out of 110 questions in our experiment, the system managed to answer eighty questions correctly, twenty-one questions incorrectly and failed to deliver an answer for the remaining nine questions. This puts the overall performance of Lemaza at, precision $P@1 = 79.21\%$, recall $R = 72.73\%$ and $c@1 = 78.68\%$. For cases where Lemaza did not deliver an answer, we noted that none of the five relevant rhetorical relations (see Section 5.4) were in the context. For the sample text titled, (الشاي الأخضر يمكن أن يعيق السرطان: *Al$Ay AlAXDr 'ymkn An yEyq AlsrTAn'*) *The Green Tea 'can hinder cancer'*, the full text is available at http://news.bbc.co.uk/hi/arabic/sci_tech/newsid_3127000/3127659.stm, our system failed to return an answer for the query, (لماذا يعد الشاي الأخضر مفيداً؟: *lmA*A yEd Al$Ay AlAXDr mfydA?*) *Why is green tea useful?*

The next experiment evaluates the impact of skipping an individual preprocessing step on the system's performance. Figure 6 shows the effect of stop-word removal, stemming and keyword expansion on the overall performance of Lemaza. The best performance is achieved when stop-words are not removed, while executing stemming and keyword expansion. As we mentioned earlier, Lemaza uses root-based stemmer (Khoja and Roger 1999). Since there are many other stemmers, so it is worth looking what impact will a different stemmer have on the performance. We picked the stemmer in Morphological Analysis and Disambiguation for Arabic (Habash and Rambow 2005; Habash, Rambow and Roth 2009). Morphological Analysis and

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Disambiguation for Arabic, a freely available toolkit that derives morphological and contextual information from raw Arabic text, and uses this information for different NLP tasks, e.g. POS tagging and stemming. Executing LEMAZA with Morphological Analysis and Disambiguation for Arabic stemmer on the 110 questions yielded the exact same performance we had with the original stemmer.

To put our system in perspective, we will compare LEMAZA's performance with two other systems, Salem *et al.* (2010) and Verberne *et al.* (2007). All three systems are RST-based and are geared toward answering *why*-questions. The first two are for questions in Arabic and the last one in English. For true performance analysis, the systems must be tested using the same dataset (corpus and *why*-question and answer pairs). Unfortunately, we are unable to report on how LEMAZA fairs against the system in Salem *et al.* (2010) as we were unable to arrange with them to share their dataset, or let them run their system on our dataset. On the other hand, the dataset in Verberne *et al.* (2007) is in English, so it is of no use for LEMAZA which works on Arabic documents. Table 5 summarizes the performance of the same three systems as reported by their respected authors. Note that LEMAZA outperformed the other two systems. Salem *et al.* (2010) tested their system on ninety-eight *why*-question and answer pairs (in Arabic) and reported fifty-four (forty-four) correctly (wrongly) answered questions, respectively. This means the measure of recall and precision are 55.1%. The authors themselves did not report precision, so in calculating the precision, we assume that all their wrong answer were actually cases of incorrect answer and there were no cases of no answer. The English *why* QA system (Verberne *et al.* 2007) tested their system using a set of 336 *why*-question and answer pairs and achieved a recall of 53.3%, precision was not reported. This low recall can be attributed to the data source used for the experiment. The author stated that the maximum recall that can be obtained from the use of RST relations and RST Treebank is fifty-eight%, however, if they filter out questions that require world knowledge, the recall jumps to 73.9%. We also have the performance measure of a Japanese *why*QA system with a reported precision $P@1 = 41.8\%$ (Oh *et al.* 2013).

Finally, we conducted a test using the same dataset in QArabPro (Akour *et al.* 2011), namely the comprehension tests from Wikipedia (Ar) and the same set of forty-five *why*-question and answer pairs. The QArabPro is an Arabic QA system that handled all types of questions including *why*-questions. For QArabPro, the reported accuracy was 62.22% for the *why*-questions, while LEMAZA's accuracy was seventy-five% for that particular dataset and question–answer pairs.

7 Conclusion and future work

In this paper, we described LEMAZA, a QA system that exclusively handles *why*-questions expressed in Arabic language. Most of the existing Arabic QA systems do not handle *why*-questions, as these are more difficult to handle. Our scheme to tackle *why*-QA is a process composed into four components: question analysis, document preprocessing, document/PR and answer extraction. The last component, answer extraction, employs RST to extract the answer. In case there are multiple candidate

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Table 5. *Performance comparison with other RST based why–QA systems. The measure for the other systems is as reported by their respected authors.*

	LEMAZA	Salem *et al.*	Verberne *et al.*
Supported language	Arabic	Arabic	English
No. of questions	110	98	336
Data source	OSAC	Arabic websites	RST Treebank
Document size (words)	150–1000	150–350	350–550
Recall	72.7%	55.1[a]	53.3%
Precision	79.2%	55.1[b]	–

[a]Measure covers *why* and *how to* questions.
[b]Under the assumption there were no cases of unanswered questions.

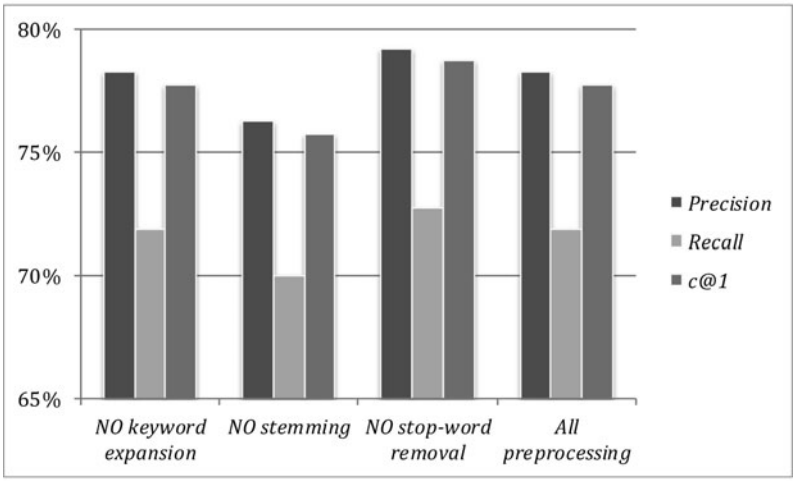


Fig. 6. The effect of eliminating an individual preprocessing step on the overall performance of LEMAZA.

answers corresponding to different rhetoric relations, then the system chooses the answer whose RST relation has the highest priority. We tested our system using 110 *why*-question and answer pairs using a corpus of 700 documents extracted from Open Source Arabic Corpora. LEMAZA's best performance measure is, recall $R = 72.73\%$, precision $P@1 = 79.21\%$ and $c@1 = 78.68\%$. This was achieved when stop-words were retained.

For future work, we plan to increase the number of questions in the test collection using a larger corpus. We also plan to improve our PR component so to use the web for searching for an answer when it fails to find one in the corpus.

References

Abouenour, L., Bouzouba, K., and Rosso, P. 2013. An evaluated semantic query expansion and structure-based approach for enhancing Arabic

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

question/answering. *International Journal on Information and Communication Technologies (IJICT)* **3**(3): 37–51.

Abouenour, L., Bouzoubaa, K., and Rosso, P. 2008. Improving Q/A using Arabic wordnet. In *Proceedings of the 2008 International Arab Conference on Information Technology (ACIT'08)*, Tunisia.

Akour, M., Abufardeh, S., Magel, K., and Al-Radaideh, Q. 2011. QArabPro: a rule based question answering system for reading comprehension tests in Arabic. *American Journal of Applied Sciences* **8**(6): 652–61.

Al-Kabi, M. N., Kazakzeh, S. A., Abu Ata, B. M., Al-Rababah, S. A., and Alsmadi, I. M., 2015. A novel root based Arabic stemmer. *Journal of King Saud University – Computer and Information Sciences* **27**(2): 94–103.

Al-Sanie, W. 2005. *Towards an Infrastructure for Arabic Text Summarization using Rhetorical Structure Theory*. Master's Thesis, King Saud University, Riyadh, Saudi Arabia.

Asher, N., and Lascarides, A. 2003. *Logics of Conversation*. Cambridge: Cambridge University Press.

Azmi, A. M., and Al-Thanyyan, S. 2012. A text summarizer for Arabic. *Computer Speech and Language* **26**(4): 260–73.

Azmi, A. M., and Aljafari, E. A. 2017. Universal web accessibility and the challenge to integrate informal Arabic users: a case study. In *Universal Access in the Information Society (UAIS)*, Springer, doi:10.1007/s10209-017-0522-3.

Azmi, A. M., and Almajed, R. S. 2015. A survey of automatic Arabic diacritization techniques. *Natural Language Engineering (NLE)* **21**(3): 477–95.

Azmi, A. M., and AlShenaifi, N. 2014. Handling 'why' questions in Arabic. In *Proceedings of the 5th International Conference on Arabic Language Processing (CITALA '14)*, Oujda, Morocco. Available at `http://www.citala.org/papers/paper_56.pdf`.

Bateman, J., and Delin, J. 2006. Rhetorical structure theory. In K. Brown (ed.), *Encyclopedia of Language and Linguistics*, 2nd ed., pp. 589–97. Amsterdam: Elsevier, BV.

Benajiba, Y. 2007. *Arabic Question Answering*. Master's Thesis, Universidad Politécnica de Valencia, Spain.

Benajiba, Y., Rosso, P., and Soriano, J. 2007. Adapting the JIRS passage retrieval system to the Arabic language. In *Computational Linguistics and Intelligent Text Processing*, pp. 530–41. Lecture Notes in Computer Science, vol. 4394. Berlin Heidelberg: Springer.

Bosma, W. 2005. Extending answers using discourse structure. In *RANLP 2005 Workshop on Crossing Barriers in Text Summarization Research*, Borovets, Bulgaria.

Brini, W., Ellouze, M., Trigui, O., Mesfar, S., Belguith, L. H., and Rosso, P. 2009. Factoid and definitional Arabic question answering system. In *NOOJ '09*, Tozeur, Tunisia, pp. 243–55.

El-Khair, I. A. 2006. Effects of stop words elmination for Arabic information retrieval: a comparative study. *International Journal of Computing and Information Sciences* **4**(3): 119–33.

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Ezzeldin, A. M., and Shaheen, M. 2012. A survey of Arabic question answering: challenges, tasks, approaches, tools, and future trends. In *Proceedings of the 13th International Arab Conference on Information Technology (ACIT'12)*, pp. 280–7.

Farghaly, A., and Shaalan, K. 2009. Arabic natural language processing: challenges and solutions. *ACM Transaction on Asian Language Information Processing* **8**(4): 1–22.

Ferguson, C. A. 1959. Diglossia. *Word* **15**(2): 325–40.

Gaizauskas, R., and Humphreys, K. 2000. A combined IR/NLP approach to question answering against large text collections. In *Proceedings of RIAO 2000: Content-Based Multimedia Information Access*, Paris, France, pp. 1288–1304.

Habash, N., and Rambow, O. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings 43rd Annual Meeting on Association for Computational Linguistics*, pp. 573–80.

Habash, N., Rambow, O., and Roth, R. 2009. MADA+TOKAN: a toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *Proceedings 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, Cairo, Egypt, pp. 102–9.

Hammo, B., Abu-Salem, H., Lytinen, S., and Evens, M. 2002. QARAB: a question answering system to support the Arabic language. In *Workshop on Computational Approaches to Semitic Languages (ACL '02)*. Association for Computational Linguistics, pp. 55–68.

Hammo, B., Abuleil, S., Lytinen, S., and Evens, M. 2004. Experimenting with a question answering system for the Arabic language. *Computers and the Humanities* **38**(4): 397–415.

Higashinaka, R., and Isozaki, H. 2008. Corpusbased question answering for why questions. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP 2008)*, Hyderabad, India, pp. 419–25.

Iruskieta, M., da Cunha, I., and Taboada, M. 2014. A qualitative comparison method for rhetorical structures: identifying different discourse structures in multilingual corpora. *Language Resources & Evaluation* **49**(2): 263–309.

Kanaan, G., Hammouri, A., Al-Shalabi, R., and Swalha, M. 2009. A new question answering system for the Arabic language. *American Journal of Applied Sciences* **6**(4): 797–805.

Keskes, I., Zitoune, F. B., and Belguith, L. H. 2014. Splitting Arabic texts into elementary discourse units. *ACM Transaction Asian Language Information Processing* **13**(2): 9:19:23.

Khoja, S., and Roger, G. 1999. Stemming Arabic text. Technical Report, Computing department, Lancaster University.

Larkey, L. S., Ballesteros, L., and Connell, M. E. 2002. Improving stemming for Arabic information retrieval: light stemming and cooccurrence analysis. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, pp. 275–82.

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Mann, W. C., and Thompson, S. A. 1988. Rhetorical structure theory: toward a functional theory of text organization. *Text* **8**(3), 243–81.

Manning, C. D., Raghavan, P., and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.

Marcu, D. 1997. *The Rhetorical Parsing, Summarization, and Generation of Natural Languag Texts*. PhD's Thesis, University of Toronto, Toronto, Canada.

Marcu, D. 1998. Improving summarization through rhetorical parsing tuning. In *Proceedings of the 6th Workshop on Very Large Corpora*, Montreal QC, Canada, pp. 206–15.

Marcu, D. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. Cambridge, MA: MIT Press.

Nakov, P., Màrquez, L., Magdy, W., Moschitti, A., Glass, J., and Randeree, B. 2015. Semeval-2015 task 3: answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval '15)*, Denver, Colorado, pp. 269–81.

Nakov, P., Màrquez, L., Moschitti, A., Magdy, W., Mubarak, H., Freihat, A., Glass, J., and Randeree, B. 2016. SemEval- 2016 task 3: community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval '16)*, San Diego, California.

Nakov, P., Hoogeveen, D., Màrquez, L., Moschitti, A., Mubarak, H., Baldwin, T., and Verspoor, K. 2017. SemEval- 2017 task 3: community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval '17)*, Vancouver, Canada.

Oh, J. H., Torisawa, K., Hashimoto, C., Kawada, T., De Saeger, S., Kazama, J., and Wang, Y. 2012. Why-question answering using sentiment analysis and word classes. In *Proceedings 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jeju Island, Korea, pp. 368–78.

Oh, J. H., Torisawa, K., Hashimoto, C., Sano, M., De Saeger, S., and Ohtake, K. 2013. Why-question answering using intra and intersentential causal relations. In *Proceedings 51st Annual Meeting of the Association for Computational Linguistic (ACL 2013)*, Sofia, Bulgaria, pp. 1733–43.

Peñas, A., and Rodrigo, A. 2011. A simple measure to assess non-response. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL–HLT '11)*, Portland, Oregon, pp. 1415–24.

Peñas, A., Hovy, E. H., Forner, P., Rodrigo, Á, Sutcliffe, R. F. E., Sporleder, C., Forascu, C., Benajiba, Y., and Osenova, P. 2012. Overview of QA4MRE at CLEF 2012: question answering for machine reading evaluation. In *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes*, Rome, Italy.

Rosso, P., Benajiba, Y., and Lyhyaoui, A. 2006. Towards an Arabic question answering system. In *Proceedings of the 4th Conference on Scientific Research Outlook & Technology Development in the Arab World*, Syria, pp. 11–14.

Ryding, K. C. 2005. *A Reference Grammar of Modern Standard Arabic*. Cambridge: Cambridge University Press.

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Saad, M. K., and Ashour, W. Nov. 2010. OSAC: open source Arabic corpora. In *Proceedings of the 6th International Conference on Electrical and Computer Science (EECS'10)*, Lefke, North Cyprus, pp. 118–23.

Salem, Z., Sadek, J., Chakkour, F., and Haskkour, N. 2010. Automatically finding answers to 'Why' and 'How to' questions for arabic language. In R. Setchi, I. Jordanov, R. Howlett, and J. Lakhmi (eds.), *Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 6279, pp. 586–93. Lecture Notes in Computer Science. Berlin Heidelberg: Springer.

Salton, G., Wong, A., and Yang, C. S. 1975. A vector space model for automatic indexing. *Communications of ACM* **18**(11): 613–20.

Scott, D. R., and de Souza, C. S. 1990. Getting the message across in RST-based text generation. In R. Dale, C. Mellish, and M. Zock (eds.), *Current Research in Natural Language Generation*, pp. 47–73. San Diego CA: Academic Press Professional Inc.

Seif, A., Mathkour, H., and Touir, A. 2005. An RST computational tool for the Arabic language. In *Proceedings of the 7th International Conference on Information Integrationed Web-based Applications Services (iiWAS'05)*, Kuala Lumpur, Malaysia, pp. 527–34.

Semmar, N., Laib, M., and Fluhr, C. 2006. Using stemming in morphological analysis to improve Arabic information retrieval, *Traitement automatique des Langues naturelles (TALN 2006)*, Leuven, Belgium, pp. 317–26.

Severyn, A., and Moschitti, A. 2012. Structural relationships for largescale learning of answer reranking. In *Proceedings of the 35th Annual ACM SIGIR Conference (SIGIR 2012)*, Portland, Oregon, pp. 741–50.

Severyn, A., and Moschitti, A. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th Annual ACM SIGIR Conference (SIGIR 2015)*, Santiago, Chile, pp. 373–82.

Shaheen, M., and Ezzeldin, A. M. 2014. Arabic question answering: systems, resources, tools, and future trends. *Arabian Journal for Science and Engineering* **39**(6): 4541–64.

Silberztein, M. 2005. NooJ: a linguistic annotation system for corpus processing. In *Proceedings of the Conference on Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, Vancouver BC, Canada.

Taboada, M., and Stede, M. 2009. Introduction to RST (Rhetorical Structure Theory). Slides available at `http://edu.cs.uni-magdeburg.de/EC/lehre/wintersemester-2011-2012/dokumentverarbeitung/folien-und-materialien/RST_Introduction.pdf`

Trigui, O., Belguith, L. H., and Rosso, P. 2010. DefArabicQA: Arabic definition question answering system. In *Proceedings of the 7th LREC Workshop on Language Resources and Human Language Technologies for Semitic Languages*, Valletta, Malta, pp. 40–5.

Tymoshenko, K., and Moschitti, A. 2015. Assessing the impact of syntactic and semantic structures for answer passages reranking. In *Proceedings of The 24th ACM International Conference on Information and Knowledge Management (CIKM 2015)*, Melbourne, Australia, pp. 1451–60.

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press

Verberne, S. 2010. *In Search of the Why*. PhD Thesis, University of Nijmegen, The Netherlands.

Verberne, S., Boves, L., Coppen, P.-A., and Oostdijk, N. 2007. Discourse-based answering of why-questions. *Traitement automatique des Langues (TAL)*, Published by Association pour le traitement automatique des langues (ATALA), Paris France **47**(2): 21–41.

Verberne, S., Boves, L., Oostdijk, N., and Coppen, P.-A. 2010. What is not in the bag of words for Why-QA? *Computational Linguistics* **36**(2): 229–45.

Verberne, S., van Halteren, H., Theijssen, D., Raaijmakers, S., and Boves, L. 2011. Learning to rank for why-question answering. *Information Retrieval* **14**(2): 107–32.

Webber, B. 2004. D-LTAG: extending lexicalized TAG to discourse. *Cognitive Science* **28**(5): 751–79.

Zhao, Y.-M., Xu, Z.-M., Guan, Y., and Wang, X.-L. 2006. An open domain question answering system based on improved system similarity model. In *Proceedings of the 5th International Conference on Machine Learning and Cybernetics*, Dalian, China, pp. 4521–6.

https://doi.org/10.1017/S1351324917000304 Published online by Cambridge University Press