Natural Language Engineering **25** (1): 5–41. © Cambridge University Press 2018
doi:10.1017/S1351324918000426

*Arabic community question answering**

PRESLAV NAKOV[1], LLUÍS MÀRQUEZ[2]†,
ALESSANDRO MOSCHITTI[3]†
and HAMDY MUBARAK[1]

[1]*Arabic Language Technologies, Qatar Computing Research Institute, HBKU, HBKU Research Complex, PO box 5825, Doha, Qatar*
e-mails: pnakov@hbku.edu.qa, hmubarak@hbku.edu.qa
[2]*Amazon, Carrer de Tànger 76, 08018, Barcelona, Spain*
e-mail: lluismv@amazon.com
[3]*Amazon, 1240 Rosecrans Ave #120, Manhattan beach, CA 90266, USA*
e-mail: amosch@amazon.com

(*Received 19 July 2017; revised 19 September 2018; accepted 21 September 2018*)

Abstract

We analyze resources and models for Arabic community Question Answering (cQA). In particular, we focus on CQA-MD, our cQA corpus for Arabic in the domain of medical forums. We describe the corpus and the main challenges it poses due to its mix of informal and formal language, and of different Arabic dialects, as well as due to its medical nature. We further present a shared task on cQA at SemEval, the International Workshop on Semantic Evaluation, based on this corpus. We discuss the features and the machine learning approaches used by the teams who participated in the task, with focus on the models that exploit syntactic information using convolutional tree kernels and neural word embeddings. We further analyze and extend the outcome of the SemEval challenge by training a meta-classifier combining the output of several systems. This allows us to compare different features and different learning algorithms in an indirect way. Finally, we analyze the most frequent errors common to all approaches, categorizing them into prototypical cases, and zooming into the way syntactic information in tree kernel approaches can help solve some of the most difficult cases. We believe that our analysis and the lessons learned from the process of corpus creation as well as from the shared task analysis will be helpful for future research on Arabic cQA.

1 Introduction

Question answering forums such as Quora, StackOverflow, and Yahoo! Answers allow communities of users to discuss problems around concrete topics, and to find solutions and answers using collective knowledge. Automatically searching for existing related questions or existing posts that directly answer a newly posed

*The authors would like to thank the anonymous reviewers for their constructive comments, which have helped us improve the quality of the paper. This research was performed by the Arabic Language Technologies (ALT) group at the Qatar Computing Research Institute (QCRI), HBKU, part of Qatar Foundation. It is part of the Interactive Systems for Answer Search (Iyas) project, which is developed in collaboration with MIT-CSAIL.
†Work conducted while these authors were at QCRI and HBKU.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

question are challenging tasks that have attracted a lot of attention from the Natural Language Processing (NLP) and Information Retrieval (IR) communities (Qiu and Huang 2015; dos Santos *et al.* 2015; Severyn and Moschitti 2015; Shen *et al.* 2015; Wang and Nyberg 2015; Zhou *et al.* 2015; Lei *et al.* 2016).

Apart from the difficulties inherent to the type of language used in open forums (ungrammatical, informal, error-prone, etc.), another challenge is the fact that the most helpful answers to a user question may be buried in long discussion threads associated with one or more similar questions. Moreover, both the questions and the answers can use open language, referring to knowledge that goes beyond traditional *factual* question answering.

Recently, some English resources have been developed for the NLP community that have helped researchers to train new systems and to compare results under fair experimental conditions (Hoogeveen, Verspoor and Baldwin 2015; Lei *et al.* 2016), notably as part of SemEval, the International Workshop on Semantic Evaluation (Nakov *et al.* 2015, 2016b, 2017a). For Arabic, the existing resources and systems for community question answering (cQA) are rather limited. A notable exception is the dataset on medical forums from the SemEval-2016 Task 3 on cQA,[1] which we describe in detail in this article. Apart from the lack of resources, Arabic cQA has some specific characteristics that make it particularly difficult, e.g., the lack of capitalization in the Arabic script and the use of many dialects in addition to Modern Standard Arabic (MSA) (see Section 3.2 for details).

Here, we present and analyze resources and advanced models for enabling better Arabic cQA. We first describe CQA-MD, a cQA corpus for Arabic in the domain of medical forums. Medical websites attract many users in the Arab world, but the medical domain has been largely ignored in the Arabic question answering literature. The creation of this corpus involved many steps: from crawling and processing the data from the medical websites, to retrieving appropriate question–answer (QA) passage pairs, to annotating them using crowdsourcing to produce training and testing datasets, and finally to documenting and making the corpus available.

In order to promote the Arabic corpus and the exploration of Arabic medical cQA in general, we ran an international evaluation challenge within the framework of SemEval-2016. This is the above-mentioned Task 3 on cQA (Nakov *et al.* 2016b). In this article, we describe the most interesting systems, the results and the lessons learned from that evaluation exercise. In doing so, we focus on the two best-performing systems, SLS (Mohtarami *et al.* 2016) and ConvKN (Barrón-Cedeño *et al.* 2016), which present different, complementary solutions both interesting from a modeling and a linguistic viewpoint. SLS implements convolutional and recurrent neural networks relying on textual embeddings, while ConvKN is a kernel-based classifier that uses tree kernels (TKs) (Moschitti 2006) to inject syntactic and semantic information to compute the similarity between questions and answers.

Additionally, we perform three new studies in order to better understand the advantages and limitations of current Arabic cQA systems. First, we experimented with combining the outputs of all submissions from all participating teams (up to

[1] http://alt.qcri.org/semeval2016/task3/

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

three per team), obtaining substantially better results, i.e., showing that the systems produce complementary errors, and thus there is a large room for improvement in the task. Second, we perform manual analysis of the examples that were difficult to rank for all systems in the competition. We categorize the examples into some prototypical cases, based on which we give general suggestions about how to improve Arabic cQA in the future. Third, we analyze the impact of the syntactic information used by ConvKN, by comparing the output of ConvKN to a version of the same system without TKs. This shows how syntactic TKs can help solving some complex cases.

The rest of the article is organized as follows: Section 2 presents related work on Arabic question answering and on cQA in general. Section 3 describes the CQA-MD Arabic corpus for cQA, which we created. Section 4 presents the shared task we organized on Arabic cQA using CQA-MD, and discusses the systems, the approaches, and the results. Section 5 contains further evaluation, which explores the potential of combining the outputs of the participating systems. Section 6 summarizes the manual analysis done on the systems' output: we describe the typology of the most frequent errors, and also the contribution of the syntactic information in the TK approach. Finally, Section 7 concludes and discusses some potential lines for future research.

2 Related work

QA systems heavily use techniques from NLP, as well as a lot of resources. The scarceness of the latter has limited the development of QA systems for Arabic, e.g., compared to English. This aspect is illustrated well by Ezzeldin and Shaheen (2012) and Ray and Shaalan (2016), who conducted surveys of Arabic QA, exploring the challenges, the tasks, the approaches, and the tools. In addition to the scarceness of language resources, corpora, and benchmarks, they pointed out to some Arabic-specific aspects such as its complex and rich morphology, the ambiguity due to the omission of diacritics, and the lack of capitalization for named entities. These aspects highly affect all three basic components of a QA system: question analysis, passage retrieval, and answer extraction. In the following, we report on state-of-the-art QA systems, which provide solutions with respect to the above components.

Question Analysis. Rosso *et al.* (2005) analyzed Arabic questions by removing the stop words, extracting the named entities, and classifying the questions in four categories: *Name*, *Date*, *Quantity*, or *Definition*. Subsequent work further extracted question focus (the main noun phrase), and important noun phrases and named entities (Wissal *et al.* 2009; Abdelbaki, Shaheen and Badawy 2011; Kanaan and Hammouri 2009). The latter three studies only used a small number of questions (e.g., hundred questions), which makes it hard to draw any general conclusions.

Passage Retrieval. Benajiba and Rosso (2007) ranked the retrieved passages according to the relevant question terms appearing in the passage. They assigned a higher rank to passages with a smaller distance between the keywords. Kanaan and Hammouri (2009) used stop word removal, root extraction, and term weighting, together with cosine similarity between document words and question words.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

Abouenour (2011) used a three-level approach in passage retrieval based on (i) query keywords, expanded with morphology and semantic concepts from the Arabic WordNet, (ii) *n*-gram structure to give higher weights to question words closer to each other, and (iii) semantics encoded in concept graphs for passage reranking.

Answer Extraction. Most studies customized manually defined patterns, heuristics, rules, and semantic similarity between question the focus and the retrieved answers using *n*-grams, e.g., as in (Trigui *et al.* 2010) and (Abdelbaki *et al.* 2011).

End-to-end QA Systems. Several complete systems have also been proposed in previous work. For example, QARAB (Hammo, Abu-Salem and Lytinen 2002) was one of the early systems for unstructured Arabic QA and was used for the Al-Raya newspaper[2] in Qatar; however, its evaluation was limited to only 113 factoid questions.

Al Chalabi (2015) used automatic classifiers of Arabic factoid questions for better identifying the answer to a given question. Their question taxonomy reflected question words, such as *who*, *what*, *when*, *how*, etc. They built question classifiers using regular expressions, patterns, and a context-free grammar for each interrogative particle, along with 200 training questions[3] and tested on another set of 200 questions made available by TREC and CLEF. The reported results were promising also compared to English QA.

Wissal *et al.* (2009) proposed Question-Answering System for Arabic Language. They used IR and NLP techniques for processing both factoid questions and documents. Question-Answering System for Arabic Language exploits the linguistic annotation system of Silberztein (2005) for also dealing with definitional questions.

Trigui *et al.* (2010) proposed a QA system for definitional questions focused on entities. Their system identifies the definitions of entities using Web resources.

Salem *et al.* (2010) built a system for answering *why* and *how* questions expressed in the Rhetorical Discourse Structure formalism. The system was evaluated on data extracted from Arabic websites, and demonstrated decent accuracy.

Akour *et al.* (2011) designed QArabPro, which is a rule-based system for reading comprehension tests. Although the system tackled any type of question, the dataset was limited to only seventy-five questions.

Overall, the problems with the aforementioned systems can be summarized as follows: small datasets for testing, and focus on factoid questions, not covering different question types and styles. In contrast, our work is focuses on cQA and enables for the first time (i) the use of large-scale user-created data sets for training and testing Arabic cQA systems, and (ii) studying real-world QA applications and the challenges they pose.

Community Question Answering (cQA) *for Arabic.* The above section has highlighted the limited development in Arabic QA. Regarding cQA, the situation is even

[2] Al-Raya newspaper: http://www.raya.com
[3] These were created by Benajiba, http://users.dsic.upv.es/ ybenajiba/

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

more critical, as this is a much newer research area, also requiring to deal with other challenges such as the use of informal language, typical characterizing forums, and other web-based communities.

For English, several initiatives have been carried out recently. For example, there has been a recent revival of the TREC Question Answering track at TREC 2015 and 2016, as LiveQA (Agichtein *et al.* 2015), which provided a challenge and data for answering *real user questions*, posted live to the Yahoo! Answers site. There have been also relevant workshops, e.g., at SIGIR (Moschitti *et al.* 2016). The most relevant initiative was that on cQA at SemEval-2015 and 2016, i.e., Task 3 (Nakov *et al.* 2015, 2016b), which focused on answering new questions using a cQA forum (Qatar Living). In particular, participants were asked to rerank the results returned by a search engine, and also to select the good answers in the forum. See for example the systems developed in (Barrón-Cedeño *et al.* 2015), (Joty *et al.* 2015; Nicosia *et al.* 2015; Mihaylova *et al.* 2016; Nakov, Màrquez and Guzmán 2016a; Joty, Màrquez and Nakov 2016). The challenge further proposed question–question similarity and answer selection subtasks (Barrón-Cedeño *et al.* 2016; Filice *et al.* 2016; Lei *et al.* 2016).

Regarding cQA for Arabic, we could not find any other research in addition to our SemEval effort in years 2015 and 2016. Thus, the present work both contributes to standard QA research and is foundational with respect to Arabic cQA.

First, as described in Section 3, given an input question, for Arabic we rank pairs of (question, *correct* answer). In contrast, the closest English cQA subtask at SemEval, subtask B, given an input question asked participants to rank potentially relevant questions, but without the availability of an answer. The difference is because for Arabic, we could exploit the *correct* answers provided by doctors.

Second, for English, we used Google to retrieve the candidates. This is a reasonable choice as Qatar Living, the underlying website, also uses Google to search in their forum (Hoque *et al.* 2016). In contrast, for Arabic we used a local search engine, Apache Solr, which indexed the data we crawled from the target Arabic forums as described in Section 3.1. This choice was driven by our desire to have a better control of the retrieval aspect, even though it was not part of the challenge.
However, this prevented us from exploiting several features of the Google algorithms for Arabic, e.g., their powerful language models (LMs), which could fill the lexical gap between the input question and the retrieved question/answer pairs. A direct consequence was that Solr was not able to retrieve candidates containing relevant QA pairs for many queries, even when using a large number (hundred) of retrieved candidates. To overcome this additional complexity, we needed to generate different query models: selection of medical terms, higher weights to such terms, use of stemming and lemmatization, different term weighting schemes, etc. This process was manual-labor intensive and could be seen as part of our manual selection process. In other words, the selection of the best list of candidates was not supposed to be a fully automatic process.

Note that this does not limit the validity of our findings in any way since (i) our study assumes that a powerful search engine is able to provide an initial set of good candidates and this is exactly what our semi-automatic retrieval approach can

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

generate, and (ii) all the methods of the participants' systems (and the challenge in general) were specifically detached from the retrieval part as the task consisted in reranking an initial set of candidates. Moreover, note that it is standard in the IR community to provide retrieval benchmarks, e.g., in the TREC QA task, the candidates that are eventually judged for relevance by humans are obtained by merging the top candidates returned by each of the participating systems.

3 Arabic data for cQA

In this section, we first describe the creation of the CQA-MD corpus for Arabic cQA in the medical domain, which to the best of our knowledge, is the first large and annotated Arabic resource for studying cQA problems. Then, we analyze the main characteristics of CQA-MD and the challenges of Arabic cQA in general.

3.1 CQA-MD: a medical arabic corpus for cQA

We used data from the medical domain, which is largely ignored for Arabic.[4] We extracted data from three popular Arabic medical websites that allow visitors to post questions related to health and medical conditions, and to get answers by professional doctors within days for free. In particular, we collected 1,531 QA pairs from WebTeb,[5] 69,582 pairs from Al-Tibbi,[6] and 31,714 pairs from the medical corner of Islamweb.[7]

In order to generate our collection, we used the 1,531 questions from WebTeb as our original questions, and we looked to find related QA pairs from the other two websites that can answer these questions.

We collected over 100,000 QA pairs in total from the other two websites, we indexed them in Solr,[8] and we searched them trying to find answers to the WebTeb questions.

We used several different query/document formulations to perform twenty-one retrieval runs, and we merged the retrieved results, ranking them according to the reciprocal rank fusion algorithm (Cormack, Clarke and Buettcher 2009). Finally, we truncated the resulting list to the thirty top-ranked QA pairs, which yielded 45,164 QA pairs[9] for the 1,531 original questions.

Note that the maximum number of candidates to be reranked, namely thirty, does not follow a precise rule or best practice. However, we considered two main factors that led us to select thirty candidates. First, while for the English cQA task, there is a large percentage of questions having at least one correct answer in the first ten candidates retrieved by a search engine, for our Arabic cQA dataset, this

[4] A notable exception is the previous edition of the task at SemEval-2015 (Nakov *et al.* 2015), where we used a dataset from the Fatwa website.
[5] http://www.webteb.com
[6] http://www.altibbi.com
[7] http://consult.islamweb.net
[8] https://lucene.apache.org/solr/
[9] We had less than thirty answers for some questions.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

percentage was much lower. Second, in the field of passage retrieval, which is similar to our task, it has been observed that even powerful rerankers fail to push relevant candidates to the top if they occur too low in the original retrieval list. Therefore, the number thirty comes from a trade-off between (i) the need to have at least some candidates to rerank (otherwise, the task would not be interesting and the difference between the participating systems would just depend on a small number of questions), and (ii) the awareness that too long lists would have not positively impacted the accuracy of the participating systems. In contrast, having many more candidates would have caused inefficiency for both training and testing. Although, we did not systematically study the distribution of correct questions with respect to the length of the retrieved candidates list, we practically observed that with thirty candidates, we could obtain relevant (Direct and Relevant) answers for most of our questions.

Next, we used the CrowdFlower[10] crowdsourcing annotation platform to obtain judgments about the relevance of these QA pairs with respect to the original question using the following labels:

- '**D**' (Direct): The QA pair contains a direct answer to the original question such that if the user is searching for an answer to the original question, the proposed QA pair would be satisfactory and there would be no need to search any further.
- '**R**' (Related): The QA pair contains an answer to the original question that covers some of the aspects raised in the original question, but this is not sufficient to answer it directly. With this QA pair, it would be expected that the user will continue the search to find a direct answer or more information.
- '**I**' (Irrelevant): The QA pair contains an answer that is irrelevant to the original question.

We controlled the quality of annotation using a hidden set of fifty test questions. We had three judgments per example, which we combined using the CrowdFlower mechanism, i.e., using majority voting and annotator confidence to assign the final label. The average inter-annotator agreement was eighty-one per cent.

We should note that the medical domain complicates the annotation task for Turkers, especially with respect to distinguishing between the classes 'D' and 'R', where the main difference is that the former satisfies the information need completely, while the latter does so only partially. This difference is about understanding the completeness of an answer, and thus can be often detected without a background in the medical domain. Another important factor to consider is that Turkers see the correct answer by a physician during annotation, and thus they can try to match such content. Moreover, the questions are asked by common people, and are about general medical topics, expressed in an informal and colloquial format; so, are the physicians' answers. Finally, note that ultimately we merge 'D' and 'R' for the purpose of the challenge and of our study; thus, even if these two classes happen to be confused, this would not really matter.

[10] http://www.crowdflower.com

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press



Fig. 1. (Colour online) Annotated question from the Arabic CQA-MD corpus.

Finally, we divided the data into training, development and testing datasets, based on confidence, where the examples in the test dataset were those with the highest annotation confidence. A well-trained native speaker of Arabic double-checked and manually corrected some of the annotations for the development and the testing datasets. The objective was to provide higher-quality development and testing datasets for parameter tuning and for testing, respectively.

Figure 1 shows part of the XML file we generated. It illustrates some challenges for Arabic cQA, such as long introductions, use of English words for medical terms, lack of capitalization for named entities, sloppy punctuation, spelling mistakes, etc. In the example, the original question is about a desire for pregnancy given menstrual cycle disorders and side effects for a certain medicine. The related question is about the reasons for menstrual cycle disorders in the case of taking contraceptive pills and how this can negatively affect pregnancy. In the related answer, after a long greeting, the physician explains that contraceptive pills can impact women differently, and describes some general effects of such pills on menstrual cycle and pregnancy.
We can further see that the XML file includes information about the confidence of annotation. This information is available for the training and the development datasets only, but not for the test dataset. It allows systems to trust less training examples with lower agreement.

Table 1 shows some statistics about the dataset size and the distribution of the three classes in the CQA-MD corpus.

3.2 Challenges in arabic cQA

Below, we discuss some characteristics of Arabic online content that make Arabic cQA difficult in general, and we provide some statistics and examples from our CQA-MD corpus.

- *Use of dialects*: A typical feature of Web forums is the use of dialectal Arabic instead of MSA. In many cases, dialectal words are incorrectly recognized as Out-Of-Vocabulary words by Arabic morphological analyzers, which in turn affects syntactic parsing and general language understanding as most tools

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

Table 1. *Statistics about the CQA-MD corpus*

Category	Train	Dev	Test	Total
Questions	1,031	250	250	1,531
QA pairs	30,411	7,384	7,369	45,164
– *Direct*	*917*	*70*	*65*	*1,052*
– *Related*	*17,412*	*1,446*	*1,353*	*20,211*
– *Irrelevant*	*12,082*	*5,868*	*5,951*	*23,901*
QA pairs per question				
min	12	22	25	
max	30	30	30	
mean	29.50	29.54	29.47	
mode	30	30	30	
stdev	1.07	0.88	0.77	

Original Question: بسم الله الرحمن الرحيم دكتورنا الفاضل: اود ان اسألك عن المرارة ما هي فوائدها بالجسم وماهي فوائد واضرار استئصالها الرجاء الاجابة في اسرع وقت ممكن وشكرا لكم

Corrected Question: بسم الله الرحمن الرحيم، دكتورنا الفاضل: أود أن أسألك عن المرارة، ما هي فوائدها بالجسم؟ وما هي فوائد وأضرار استئصالها؟ الرجاء الإجابة في أسرع وقت ممكن، وشكرا لكم.

Literal Translation: In the name of God the most beneficent the most merciful our moralist doctor: I would like to ask you about the bitterness what are its benefits to the body and what are the benefits and harms of its cholecystectomy please answer as soon as possible and thank you

Fig. 2. (Colour online) Challenges for question analysis: typos, punctuation, and preambles.

for Arabic processing are tailored for MSA. In particular, we calculated that in our test dataset about twenty per cent of the questions contained some dialectal words, while at the same time all answers were in MSA as they were written by professional physicians. This means that there is some mismatch between the language variety (dialect versus MSA) used in the question and in the answer, and this causes further issues when comparing the text of a question to the text of an answer. This is especially important at the syntactic level as most dialectal words are function words. However, in our data this is less important at the semantic level as terminology covering diseases and body parts are in MSA, i.e., word embeddings might be more robust with respect to dialectal variance.

- *Sloppy punctuation*: The informal, dialectal text in Web forums often neglects punctuation, unlike proper MSA text. Yet, the proper use of punctuation can be crucial for some NLP processing steps such as sentence segmentation and syntactic parsing. In particular, we considered the fact that, on average, the original questions, the related questions, and the related answers have a punctuation mark after every twelve, eighteen, and thirteen words, respectively.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

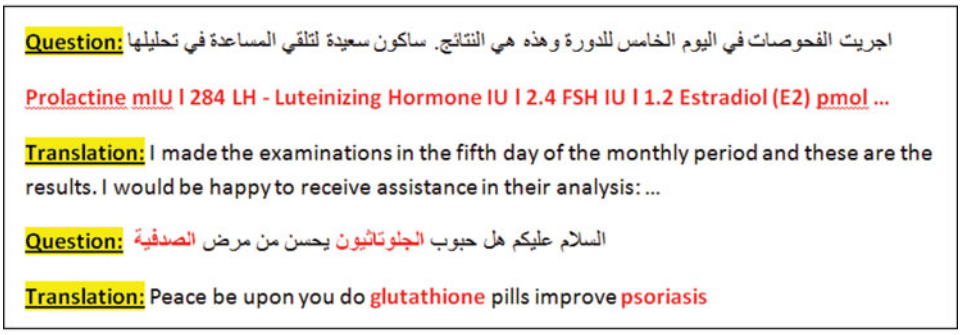


Fig. 3. (Colour online) More challenges for question analysis: English words and medical terms.

Yet, in a proper newswire text, such as a 100M-word 10-year Aljazeera[11] archive, there is a punctuation mark after seven words on the average.

- *Spelling errors*: Forum texts are written in an informal setting, and thus are not carefully edited as a newswire text would be; as a result, they often contain spelling errors. Our analysis shows that in our test dataset, eighty per cent of the questions and forty per cent of the answers had at least one typo. Most of these typos consist in omitting the hamza symbols on top and below the bare *Alef* symbol or missing the dots for the letters *ya* and *ta marbuta*, which turns them into *alef maqsura* and *ha*, respectively. These problems are somewhat easy to solve using normalization and consulting a large LM to select the best candidate in a given context; yet, this is a bit more complicated for dialectal texts as LMs are not so reliable for them. Other spelling mistakes involve missing letters, swapped letters, or concatenated/split words.
- *Long preambles and introductions*: It is common in Arabic to start a question with a religious greeting or introduction, and to end it with an acknowledgment or a prayer. Our analysis shows that fifty-eight per cent of the questions and of the answers in our dataset have such starting introductions, e.g., 'Peace be upon you and God's mercy and blessings…' and 'In the name of God the most gracious the most merciful dear brother/sister…'Also, fifty per cent of the questions and of the answers end with a salutation or a prayer such as 'and thanks', or 'May God bless/reward you'. This requires extra efforts to recognize the important parts of the questions and of the answers and potentially to ignore such introductory/ending parts.

Figure 2 shows an example that illustrates some of the difficulties described above: different types of spelling mistakes, no punctuation marks, as well as long introduction and acknowledgments that surround the core part of the question. Figure 3 shows two other cases: using English words inside Arabic questions, and complex medical terms and medicines. All these difficulties negatively affect automatic parsing and understanding.

[11] http://www.aljazeera.net

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

Below are some additional challenges we face that are specific for our particular corpus, mostly due to its medical nature:

- *Use of complex medical terms*: As our corpus is from the medical domain, it is common to find in the questions and in the answers complex medical terms (e.g., *Progesterone* in Arabic), names of medicines (e.g., *Ortho-Cyclen* in Arabic), sometimes even complex medical terms in English, e.g., *Keratoplasty* and *Zyprexa*. We found that about forty-one per cent of the questions and seventy-seven per cent of the answers contained complex medical terms (we did not consider as complex terms words such as *heart attack* and *bleeding* that are in everyday use or are generally understood by Arabic speakers), while in the remaining cases the users described their cases in a simple language; physicians also tried to simplify their way of expression in order to make their response more comprehensible.
- *Sensitive and socially embarrassing medical questions*: An important characteristic of our medical data is the abundance of questions that are arguably socially embarrassing. As people feel uneasy to talk about them face-to-face to a physician, they preferred to pose them online, hiding behind the anonymity that an Internet forum seemingly offers. In some cases, users tended to anonymously describe their cases in a lot of detail, using vocabulary that is not known to standard NLP tools, as such terms are not frequent in newswire texts that are typically used to train Arabic NLP tools. Figure 4 shows that, in our test dataset, the two most-commonly mentioned diseases, accounting for thirty-seven per cent combined, are indeed related to obstetrics and gynaecology, along with genital and urinary system diseases.[12]
- *Detailed questions and answers*: In forty-eight per cent of the questions, people were asking about general issues (e.g., what are the symptoms of a certain disease?), while in fifty-two per cent of the questions, they were asking about very specific cases (e.g., 'I am xx years old, this is my medical history. I take this medicine and I have certain symptoms, etc.'). Because users want to write everything in one question without knowing exactly what the important details from a physician's perspective are, they describe their cases with a lot of detail in terms of numbers, treatments, medical history, etc. Also, physicians reply by giving the medical description and analysis of their cases, with general and specific instructions and prescriptions. Some questions are answered by more than one physician, when the question is related to more than one medical specialty (e.g., dermatology and psychology). This makes questions and answers very long and adds extra complexity to the linguistic analysis, but also for manual annotation of the gold labels, as long questions and answers are difficult to understand. For example, the average lengths of the questions and of the answers are 40 and 119 words, respectively.

[12] We randomly selected 1,000 QA pairs from the test dataset and we manually assigned them to one or more of the twenty-seven diseases listed in the Arabic Health Encyclopedia: http://www.kaahe.org/en/

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

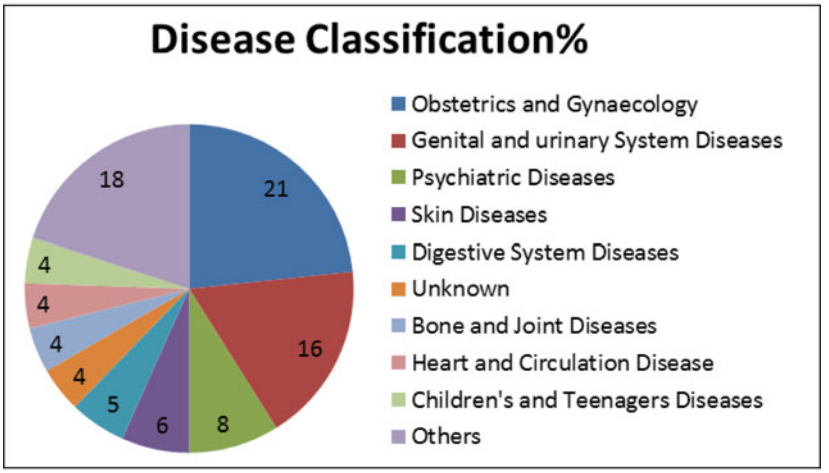


Fig. 4. (Colour online) The most frequently mentioned diseases in our test dataset.

- *Short answers*: For some questions, after a long description and lot of detail about a patient's case, we have short answers like: 'Yes', 'This is normal', or 'For two years', etc. Sometimes, a physician's reply is not a real answer to the question like: 'Send more details', 'This question was answered before', 'Unclear question', or 'Consult your doctor'. These kinds of short answers with almost no lexical overlap with the questions are challenging for systems that try to learn from data and to find syntactic or semantic relations between questions and answers. We think of annotating these answers differently in the future.

4 SemEval-2016 shared task on arabic cQA

In this section, we describe the SemEval-2016 shared task on cQA, which we organized. The task used CQA-MD as the Arabic dataset (and it also included some English subtasks). Five systems, SLS, ConvKN, Research Development International (RDI), QU-IR, and UPC_USMBA, participated in the challenge with up to three submissions: one primary, and up to two contrastive. Below, we go over the task setting, the participating systems, the main features and techniques used, the evaluation results, and the lessons learned.

4.1 Task setting

The SemEval task can be defined as follows: given an original question q_o and k candidate QA pairs $\langle q_s, a_s \rangle$, $1 \leq s \leq k$ (retrieved by a search engine), the task is to rank them with respect to their relevance to q_o. In our setting, we used $k = 30$ QA pairs associated with each q_o, where a_s is the correct answer to q_s provided by a professional physician. Operationally, the task can be defined as ranking *Direct* and *Relevant* answers above the *Irrelevant* ones (see Section 3.1). Note that the former two are considered *Relevant* in terms of the official task evaluation.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

Metrics. We used Mean Average Precision (MAP) as the primary metric. MAP is a well-established measure in IR. We further report the results for two secondary ranking measures: Mean Reciprocal Rank (MRR) and Average Recall (AvgRec).

MAP is calculated over a set of query questions Q, as the mean of the average precision (AveP) values computed on all questions:

$$\text{MAP} = \frac{1}{|Q|} \sum_{q_o \in Q} AveP(q_o)$$

where $AveP(\cdot)$ is the average of the precision value obtained for the set of the top QA pairs $\langle q_s, a_s \rangle$ computed after each relevant QA pair is retrieved.

MRR is computed as follows:

$$\text{MRR} = \frac{1}{|Q|} \sum_{q_o \in Q} \frac{1}{rank(q_o)}$$

where $rank(q_o)$ is the position of the first relevant QA pair in the list of candidates for q_o.

Computed over a set of query questions Q, AvgRec is the mean of the AveR over all questions q_o in the test dataset:

$$\text{AvgRec} = \frac{1}{|Q|} \sum_{q_o \in Q} AveR(q_o)$$

where AveR is the mean of the recall values obtained for the set of top questions, computed after each QA pair is retrieved.

4.2 Descriptions of the participating systems

Many methods can be applied to solve a ranking task, which is about defining a scoring function $r : \mathcal{Q} \times \mathcal{Q} \times \mathcal{A} \rightarrow \mathbb{R}$, for ranking $\langle q_s, a_s \rangle$ pairs with respect to q_0, where $\mathcal{Q}$ and $\mathcal{A}$ are the set of questions and answers, respectively.

Almost all participating systems, i.e., *SLS*, *ConvKN*, *RDI*, and *UPC_USMBA*, modeled r as a linear function:

$$r(q_o, q_s, a_s) = \vec{w} \cdot \phi(q_o, q_s, a_s) \tag{1}$$

where $\vec{w}$ is a weight vector and $\phi(\cdot)$ is a feature vector representation of the triple. In particular, they learn $\vec{w}$ using binary classifiers, such as Support Vector Machines (SVM) (Joachims 1999), regression models such as logistic regression (Cox 1970), or ranking models such as SVM^{rank} (Joachims 2002). It should be noted that binary SVMs showed comparable results to SVM^{rank}. Note also that *QU-IR* used different approaches for building r, i.e., Multiple Additive Regression Trees and Random Forest algorithms (Friedman 2001). However, this did not seem to yield a major difference. In fact, the critical aspect was the different approaches to feature engineering.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

SLS (MIT-CSAIL)

This SLS system by Mohtarami *et al.* (2016) applied SVMs to the following concatenation of a lexical-based and of an embedding-based feature vector:

$$\phi(q_o, q_s, a_s) = [\phi_{\text{lex}}(q_o, q_s, a_s), \phi_{\text{emb}}(q_o, q_s, a_s)]$$

The feature vector $\phi_{\text{lex}}(q_o, q_s, a_s)$ is in turn a concatenation of two feature vectors

$$\phi_{\text{lex}}(q_o, q_s, a_s) = [\phi_{\text{lex}}(q_o, q_s), \phi_{\text{lex}}(q_o, a_s)]$$

where $\phi_{\text{lex}}(x, y)$ is a vector whose coordinates correspond to twenty different types of textual similarity between x and y as described in (Nicosia *et al.* 2015), including greedy string tiling (Wise 1996), longest common sub-sequence (Allison and Dix 1986), and Jaccard coefficient (Jaccard 1901) over $[1, 4]$-grams.

The vector $\phi_{\text{emb}}(q_o, q_s, a_s)$ is a concatenation of two scalar products

$$\phi_{\text{emb}}(q_o, q_s, a_s) = [v(q_o) \cdot v(q_s), v(q_o) \cdot v(a_s)]$$

where the vectors $v(q_o)$, $v(q_s)$, and $v(a_s)$ are embedding representations of q_o, q_s, and a_s, respectively; these question/answer embeddings are the average of the embeddings of the words they are made of

$$v(X) = \sum_{w \in X} w2v(w)$$

where w is a word-token from the text of X, and $w2v(w)$ maps w to a word embedding.

The *SLS* team used two different embeddings trained using word2vec (Mikolov *et al.* 2013) on: (i) a raw domain-specific corpus of 26 millions tokens provided with the SemEval task, and (ii) an out-of-domain corpus, Arabic Gigaword,[13] which includes over a billion words, and 120,000 word types after lemmatization. These two types of embeddings, together with some text shrinking approach using TextRank by Mihalcea and Tarau (2004), were used to generate the primary and the contrastive submissions of *SLS*. More details can be found in (Barrón-Cedeño *et al.* 2016; Belinkov *et al.* 2015).

Another important characteristic of the *SLS* system is that they used a sentence-level vector rather than a vector based on the entire question or answer text, i.e., they calculated the scalar products for the pair of sentence vectors from q_o, q_s, and a_s (instead of using the entire text), and they retained the highest score only. The idea is that the questions and the answers in our dataset are often rather large and noisy, and thus it makes sense to focus on the best matching pairs of sentences only.

ConvKN (Qatar Computing Research Institute)

This system by Barrón-Cedeño *et al.* (2016) combined different features with complex convolutional kernels such as TKs (Moschitti 2006). They used the following

[13] https://catalog.ldc.upenn.edu/LDC2011T11

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

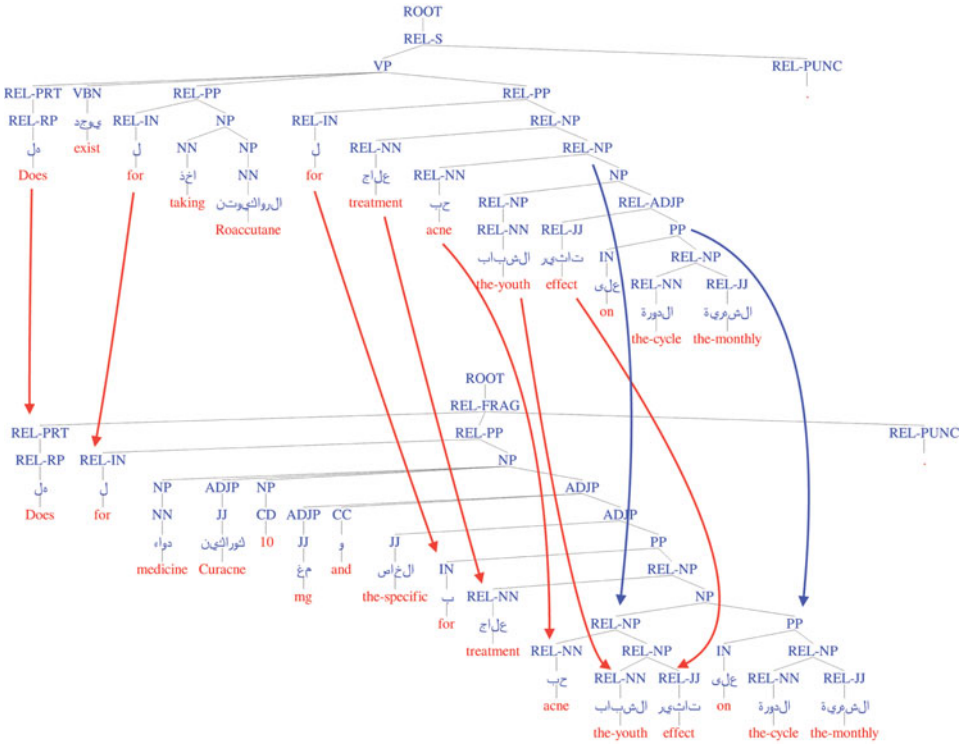


Fig. 5. (Colour online) Arabic trees of two questions connected by REL links.

representation for each $\langle q_s, a_s \rangle$ pair:

$$\phi(q_o, q_s, a_s) = \phi_{\text{lex}}(q_o, q_s, a_s) + \phi_{\text{tk}}(q_o, q_s)$$

where $\phi_{\text{lex}}(q_o, q_s, a_s)$ is the same as in *SLS*, while $\phi_{\text{tk}}(q_o, q_s)$ is a vector encoding the syntactic structure of the question pairs $\langle q_o, q_s \rangle$, e.g., as shown in Figure 5. Discussion on the use of such syntactic models can be found in Section 6.2 below.

As generating $\phi_{\text{tk}}(q_o, q_s)$ explicitly requires complex feature engineering, *ConvKN* opted for an implicit representation applying the following kernel to two general examples, $\langle q_o, q_s, a_s \rangle$ and $\langle q'_o, q'_s, a'_s \rangle$:

$$K\left(\langle q_o, q_s, a_s \rangle, \langle q'_o, q'_s, a'_s \rangle\right) = \phi(q_o, q_s, a_s) \cdot \phi(q'_o, q'_s, a'_s) =$$

$$= \phi_{\text{tk}}(q_o, q_s) \cdot \phi_{\text{tk}}(q'_o, q'_s) + \phi_{\text{lex}}(q_o, q_s, a_s) \cdot \phi_{\text{lex}}(q'_o, q'_s, a'_s)$$

where (TK is a tree kernel function)

$$\phi_{\text{tk}}(q_o, q_s) \cdot \phi_{\text{tk}}(q'_o, q'_s) = TK(t(q_o, q_s), t(q'_o, q'_s)) + TK(t(q_s, q_o), t(q'_s, q'_o))$$

This way, there was no need to extract explicit syntactic feature vectors from the text pairs. The function $t(x, y)$ is a string transformation that returns the syntactic tree from the text x, enriching it with the REL tags (see Section 6.2) computed with respect to the syntactic tree of y. *ConvKN* used the Arabic Stanford Constituency Parser (Green and Manning 2010) for generating the syntactic tree for text x and a simple annotator to mark the tree nodes with the REL tags. The latter were

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

added on the internal nodes each time a terminal node matches a word in the text y. The REL tags were then also propagated to the parent nodes (up to two levels up). See Figure 5 for an example. Finally, the *ConvKN* team used the partial TK by Moschitti (2006), which measures the similarity between trees.

RDI (Research Development International)

The *RDI* team developed a system (Magooda *et al.* 2016), which used (i) external corpora (EC) crawled from www.altibbi.com to compute the probability of $\langle q_s, a_s \rangle$ to be relevant to q_o, (ii) a LM trained on the annotated *CQA-MD* dataset; and (iii) the Arabic Wikipedia to boost the weights of medical terms.

RDI did not use supervised learning for modeling the ranking function r, and thus there is no w. Instead, they model the function r directly, which is expressed by a weighted sum of three terms as follows:

$$r(q_o, q_s, a_s) = \alpha \times \mathscr{P}\Big(\phi(q_o) \cdot \phi^{+}_{tf\times idf}(q_s, a_s), \phi(q_o) \cdot \phi^{-}_{tf\times idf}(q_s, a_s)\Big) + \\ + \beta \times \phi_{LM}(q_o) \cdot \phi_{LM}(q_s, a_s) + \gamma \times \phi_{wp}(q_o) \cdot \phi_{wp}(q_s, a_s)$$

where α, β, and γ are parameters (tuned on the development dataset), which control the impact of each of the three components.

The first component is the relevancy probability of the candidate pair $\langle q_s, a_s \rangle$ computed by the function $\mathscr{P}$[14] using the two scalar products, $\phi(q_o) \cdot \phi^{+}_{tf\times idf}(q_s, a_s)$ and $\phi(q_o) \cdot \phi^{-}_{tf\times idf}(q_s, a_s)$, where

(i) $\phi(q_o)$ is simply a binary vector representing the terms of q_o;

(ii) $\phi^{+}_{tf\times idf}$ is a TF×IDF vector of the terms in $\langle q_s, a_s \rangle$ with IDF values computed using a set of pseudo-positive QA pairs extracted from EC;[15]

(iii) $\phi^{-}_{tf\times idf}$ is a similar vector as above but using pseudo-negative pairs from EC.

The second component is the scalar product of two embedding vectors, $\phi_{LM}(q_o)$ and $\phi_{LM}(q_s, a_s)$, computed using the LM approach by Mikolov *et al.* (2011). The latter is applied to a corpus constituted by the candidate pairs, $\langle q_s, a_s \rangle$, augmented by the pairs $\langle q_o, a_s \rangle$, for each q_o and its related q_s in CQA-MD.

The third component is a scalar product between $\phi_{wp}(q_o)$ and $\phi_{wp}(q_s, a_s)$, where the vectors are constituted by n-grams, whose weights are increased if they match terms in the medical categorization system of the Arabic Wikipedia. The set of used medical terms contains about 250,000 medical subcategories extracted from the main medical category. Matching unigrams has a weight of 1.5, whereas all the other matching n-grams have a weight of 2.

[14] This is underspecified in the original paper, but it should be just a sort of odd ratio between positive and negative scores.

[15] The EC pairs are defined to be pseudo-positive if they have at least three terms in common with q_o; otherwise, they are considered to be pseudo-negative pairs.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

QU-IR (Qatar University)

This *QU-IR* system by Malhas *et al.* (2016) only used word2vec representations, $\phi_{w2v}(q_o, q_s, a_s)$, computed on (i) the concatenation of q_o and q_s, (ii) the concatenation of q_o and a_s, and (iii) the concatenation of q_o and $\langle q_s, a_s \rangle$. These representations were used to build three different systems.

Note that, unlike all other teams, *QU-IR* used these embeddings in Multiple Additive Regression Trees and Random Forest algorithms for training different non-linear ranking functions r.

UPC_USMBA (Universitat Politècnica de Catalunya)

This team used a logistic regression classifier applied to word features derived with a term extractor, and 200 similarity features extracted from the Arabic text, after processing it with the Stanford Core NLP tools (El Adlouni *et al.* 2016). They also used semantically tagged medical terms classified with the categories: Body Parts, Diseases, and Drugs.

Finally, they translated the Arabic data to English[16] to be able to apply English linguistic modules. Their final system is a combination of four classifiers: two using the Arabic text and two using the English translations.

4.3 System results

The participating teams used the training and the development data to train and to tune their models, and then applied them to the test set (the gold standard labels for which were not available to them). Each team was asked to make one primary submission (to be used for the official ranking), and was further allowed to submit one or two contrastive submissions. All teams but UPC_USMBA made three submissions.

The evaluation results are shown in Table 2. They are ranked in terms of the official evaluation measure, MAP. The first column reports the absolute rank considering both the primary and the contrastive submissions, whereas the second column shows the official system rank in the competition, which is based on the primary submissions only. The columns from the third to the seventh report the team name, the type of submission (primary or contrastive), MAP, AvgRec, and MRR, respectively. The last row in the table reports the results for a baseline that puts the candidate QA pairs in random order.

We can see that the primary submission of *SLS* achieved the highest MAP of 45.83. This suggests that word n-gram similarity and word embeddings are quite important. Indeed, the analysis of Section 4.2 shows that *SLS* used embeddings more massively and in a more clever way compared to other participating systems. For example, contrastive 1 and 2 of *ConvKN*, which only used n-gram similarities

[16] This is different from using machine translation techniques (Guzmán *et al.* 2015) to compare the question to a potential answer (Guzmán *et al.* 2016b; Wu and Zhang 2016; Guzmán, Màrquez and Nakov 2016a).

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

Table 2. *Reranking the correct question-answer pairs for a new question: results for all submissions for SemEval-2016 Task 3*

#	Rank	Team	Submission	MAP	AvgRec	MRR
1	1	SLS Mohtarami *et al.* (2016)	primary	45.83	51.01	53.66
2	2	ConvKN Barrón-Cedeño *et al.* (2016)	primary	45.50	50.13	52.55
3		SLS	contrastive1	44.94	49.72	51.58
4	3	RDI_team Magooda *et al.* (2016)	primary	43.80	47.45	49.21
5		SLS	contrastive2	42.95	47.61	49.55
6		RDI_team	contrastive1	42.18	47.03	47.93
7		ConvKN	contrastive2	39.98	43.68	46.41
8		QU-IR	contrastive2	39.07	42.72	44.14
9		RDI_team	contrastive2	38.84	42.98	42.97
10	4	QU-IR Malhas *et al.* (2016)	primary	38.63	44.10	46.27
11		ConvKN	contrastive1	38.33	42.09	43.75
12		QU-IR	contrastive1	37.80	40.96	44.39
13	5	UPC_USMBA El Adlouni *et al.* (2016)	primary	29.09	30.04	34.04
		Baseline	(random)	29.79	31.00	33.71

The first and the second columns show the rank with respect to the official MAP score of all and of the primary runs, respectively. The third and the fourth columns contain the team's name and its submission type (primary vs. contrastive). The following columns show the corresponding ranking evaluation scores. All numbers are expressed as percentages.

(syntactic TKs and word embeddings are not used in these contrastive runs), are 6–7 MAP points below the primary submission of *SLS*, with a MAP of 38.33 and 39.38. Two other important factors for the success of *SLS* are (i) the use of only the best sentence of the question or answer text (instead of the entire text), and (ii) the use of a combination of four different types of text representations, i.e., all sentence words, all word lemmata, and two other representations where the words are filtered out using TextRank with two different parameterizations. The lemma representation seems to be the most effective as alone it yields a MAP of 44.94. In contrast, TextRank seems not to produce a better outcome (at least alone) as the system using it only achieved a MAP of 42.95.

Next, *ConvKN*, which used syntactic representation was ranked second with a MAP of 45.50, i.e., only 0.33 points lower than the top system, *SLS*. This is interesting as both contrastive submissions by *ConvKN*, which did not use syntactic information, achieved a MAP of only 39.98 or 38.33. These suggests that using

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

syntactic information can yield an absolute improvement of 5.52 MAP points, which is about twelve per cent relative improvement. We speculate that if the *SLS* feature vectors were jointly used with the TK model of *ConvKN*, the resulting system would have outperformed the top system with a relatively high margin, e.g., 3–5 points.[17]

Next comes *RDI* with a MAP of 43.80, which is very interesting as this team did not use any supervised learning algorithm. Their contrastive system 1 only used the Wikipedia module, $\phi_{wp}(q_o) \cdot \phi_{wp}(q_s, a_s)$, achieving a MAP of 42.18, whereas the TF×IDF component that used an external corpus achieved a MAP of 38.84. These results suggest that the use of Wikipedia terms and additional external data can be used to boost both *SLS* and *ConvKN*,[18] which did not use such resources.[19]

Next, *QU-IR* used basic embeddings, and thus may be compared with the contrastive results of *ConvKN*, which did not use any embeddings but basic similarity features only. The best MAP of *QU-IR* is 39.97, whereas the best of *ConvKN* with no syntactic information is 39.98. This observation may be useful to estimate the impact of using a different class of learning algorithms such as Multiple Additive Regression Trees or Random Forest algorithms. We note that the latter do not yield a boost with respect to the linear models used by *ConvKN*, i.e., linear models seem to be sufficient to get state-of-the-art results.

Finally, *UPC_USMBA* used many features extracted using the Stanford Core NLP.[20] Probably, the use of such a toolkit for Arabic language was not straightforward as the resulting system achieved a very low MAP of 29.09, which is lower than the random baseline of 29.79.

Note that the system rank would not change if we did the ranking not based on MAP but on some of the other two measures, MRR and AvgRec, which suggests that the impact of the features and models proposed by the participants is rather stable, i.e., a better system is built with respect to different viewpoints.

Another interesting note is the larger difference between *SLS* and *ConvKN*, which confirms that using syntactic information helps in solving more different cases (achieving a better MAP). In contrast, when word models are effective (for at least one candidate question in the rank they probably are), they are enough to rank the best QA pair on top. This is also suggested by the *QU-IR* result, which only used embeddings, and achieved an MRR much closer to that of the preceding system.

[17] We partially and indirectly support this claim by combining the output of the primary submissions of *SLS* and *ConvKN*, using the method described in Section 5.2. This yielded a MAP score of 48.14, which constitutes 2.5+ MAP points of absolute improvement over *SLS*.

[18] Again, we tried to indirectly support this claim by combining the output of the primary submissions of *RDI* with *SLS* and also with *ConvKN*, using the method described in Section 5.2. However, this time the improvements were much smaller: from 45.83 to 45.89 for *SLS*, and from 45.50 to 46.16 for *ConvKN*.

[19] *SLS* used the Arabic Gigaword, but the external data used by *RDI* is from one of the forums we used to build our SemEval data.

[20] http://stanfordnlp.github.io/CoreNLP/

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

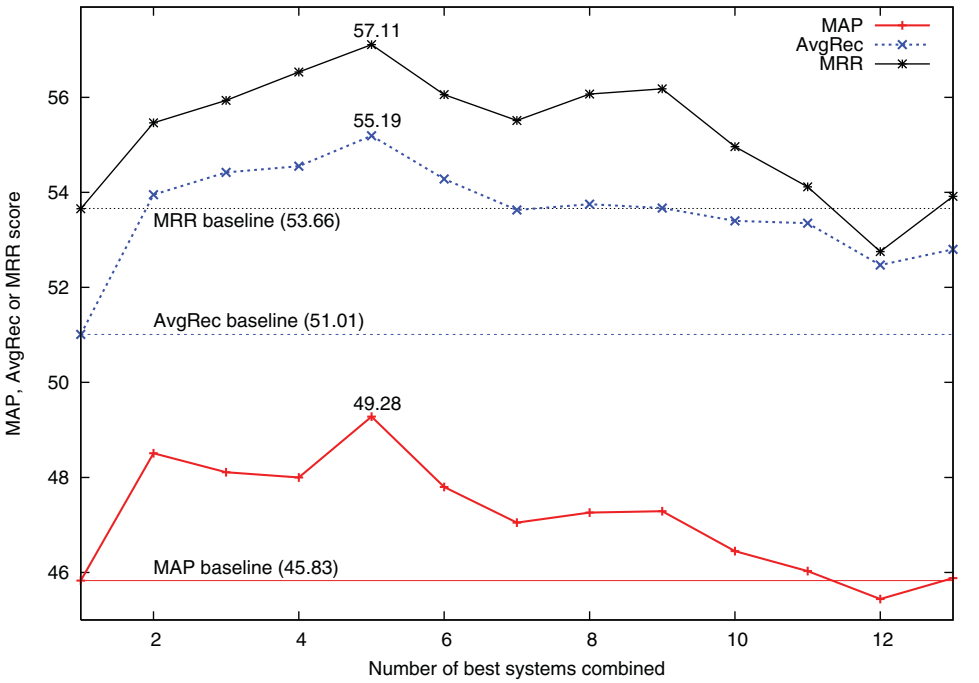


Fig. 6. (Colour online) MAP, MRR, and AvgRec for the simple combination of thirteen systems outputs.

5 Studying system combinations

In this section, we experiment with combining the outputs of the thirteen submitted runs, in order to see what further improvement can be achieved by combining the strengths of several systems that use different features and different learning methods. The idea is that if the individual runs are complementary to some extent, their errors would be relatively uncorrelated, and thus we should be able to improve over the best individual system.

5.1 Measuring the potential of system combination

In this first experiment, we combine the first n systems, $n \in \{2, 3, \ldots, 13\}$, as ordered in Table 2, in a meta-classifier. The combination approach is very simple: given an original question q_o, we compute a score $s(q_o, q_s, a_s)$ for each QA pair $\langle q_s, a_s \rangle$, and then we sum these scores for the n systems being combined as follows:

$$r(q_o, q_s, a_s) = \sum_{i=1}^{n} s_i(q_o, q_s, a_s)$$

where $s_i(q_o, q_s, a_s)$ is the score for the ith system in the combination.

We experimented with three versions of $s_i(q_o, q_s, a_s)$: (i) reciprocal rank, (ii) arithmetic rank, and (iii) min–max normalized score.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

The reciprocal rank score is defined as follows:

$$s_i^{\text{rr}}(q_o, q_s, a_s) = \frac{1}{\text{rank}_i(q_o, q_s, a_s)}$$

where $\text{rank}_i(q_o, q_s, a_s)$ is the rank that the ith system assigned to the QA pair $\langle q_s, a_s \rangle$ with respect to the original question q_o.

We also tried a scoring function based on *arithmetic ranks*, which do not decrease as fast as the reciprocal rank:

$$s_i^{\text{ar}}(q_o, q_s, a_s) = 1 - \frac{\text{rank}_i(q_o, q_s, a_s) - 1}{30}$$

but it did not improve over using the reciprocal rank.

We further tried a scoring function based on the actual numerical score, after min–max normalization:

$$s_i^{\text{sum}}(q_o, q_s, a_s) = \frac{r_i(q_o, q_s, a_s) - \min}{\max - \min}$$

where $r_i(q_o, q_s, a_s)$ is the score that the ith system gave to the QA $\langle q_s, a_s \rangle$, and min and max are the minimum and the maximum value of the scores it produced for all pairs $\langle q_s, a_s \rangle$ with respect to the original question q_o.

Using this latter score yielded slightly worse results, probably due to the different systems operating on different scales when providing the real-valued scores, which the min–max normalization could not quite fix. In contrast, using the reciprocal ranks works as a simple scale-invariant calibration of the scores of the individual systems (Severyn *et al.* 2014).

The results for this system combination using the sum of reciprocal ranks are presented in Figure 6. On the x-axis, we show the number of systems combined, and on the y-axis, we show MAP, AvgRec, and MRR scores. We also display, as the baselines for the comparison, the scores of the best individual system on each evaluation measure.

We can see that the best results are achieved when combining the top-five systems, where the improvements over the best individual system are sizable: over three points in terms of MAP and MRR, and over four AvgRec points, absolute. In most cases, the combined system performed better than the best individual system, which shows that there is clearly room for further improvement for the individual systems.

Note that this combination uses information about which system performs best in the test set when deciding which n systems to combine. Of course, this information would not be available in a real combination scenario. Consequently, we are not claiming that the best results from Figure 6 can generalize to unseen datasets. This experiment merely allowed us to show that the outputs of the systems are uncorrelated to the extent of allowing sizable improvements when combined appropriately. The next subsection deals with learning a more realistic meta-classifier.

5.2 Tuned combination

The systems that participated in the competition are obviously not equally good, and thus it might be a good idea to learn different weights for different systems.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

Table 3. *Results from the tuned system combination experiments*

Combination	MAP	AvgRec	MRR
Best individual system	45.83	51.01	53.66
Uniform combination	45.88 (+0.05)	52.80 (+1.79)	53.92 (+0.26)
Tuned: 3×13	48.47 (+2.64)	54.70 (+3.69)	55.29 (+1.63)
Tuned: 3×13 + 3 sums	48.65 (+2.82)	54.51 (+3.50)	55.34 (+1.63)

Therefore, we performed another experiment where: (i) we combined *all* thirteen systems rather than just the top n, (ii) we weighted the relative contributions of each system (we learned the best weights automatically using cross-validation), and (iii) we used both the system's scores and the corresponding ranks as features (based on the three scores from the previous section).

In particular, we trained a meta-classifier with an SVM-based learning-to-rank algorithm (SVM-rank), which, given two QA pairs that answer the same question, predicts which pair is a better answer, i.e., should be ranked higher. We train and evaluate the model using five-fold cross-validation on the test set. As we do not have access to the individual systems that participated in the competition, we could not generate predictions for the training or for the development datasets. Thus, we opted for cross-validation on the testing dataset, which allowed us to compare our results to those of the participating systems, as well as to the results in the previous experiment. We extracted and used three features from each of the thirteen systems we were combining, namely the three scores from above $s_i^{\text{rr}}(q_o, q_s, a_s)$, $s_i^{\text{ar}}(q_o, q_s, a_s)$, and $s_i^{\text{sum}}(q_o, q_s, a_s)$ for each system i. This yields $13 \times 3 = 39$ features, to which we further add three more features, namely the sums of each of the three scores, calculated over all thirteen systems that are being combined.

The results are shown in Table 3. The first two lines show baselines: the best individual system, and the simple uniform combination for all thirteen systems from before. The following lines show the results for the tuned combination using (a) the 3×13 features only, and (b) the 3×13 features plus the three sums. We can see 2.5+ points of absolute improvement in terms of MAP, 3.5+ in terms of AvgRec, and 1.5+ in terms of MRR (the three extra sums seem to have minor effect overall).

Overall, the results of the above two experiments show that the systems provide uncorrelated scores and rankings, which are in turn based on potentially uncorrelated features and learning algorithms, and thus that there is a large space for further improvement, i.e., the task is far from being solved.

6 Further analysis

In this section, we take a closer look to the output of the systems that participated in SemEval-2016 subtask D. The aim is two-fold: (i) we want to have some more qualitative information about the capabilities and the limitations of the state-of-the-art systems, and (ii) we want to learn important tips for improving them, thus enabling better cQA for Arabic in the future.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

The section is divided in two parts. The first part contains manual analysis of the most challenging examples for the participating systems, i.e., examples that were ranked wrongly by most systems. The second part is manual analysis studying the usefulness of syntactic information for the task. In this case, we analyze and compare the results of our ConvKN system (Barrón-Cedeño *et al.* 2016), which uses TKs to compare the original question with the related question–comment pairs, to the other systems that participated in the challenge.

6.1 Analysis of difficult cases

We manually analyzed some examples that were especially hard for the individual systems. In order to select the examples, we divided the test cases into *positive* (i.e., *Direct* or *Related* QA pairs, according to the gold standard annotation) and *negative* (i.e., *Irrelevant* QA pairs, according to the gold standard). We then sorted all test examples in the positive and in the negative sets by a measure of *difficulty*, and a native speaker of Arabic who has strong linguistic background studied the top twenty examples in each list. The difficulty measure for a particular example is simply the number of systems, from the set of thirteen in Table 2, that got it wrong. Note that this is not a classification task but a ranking task. Thus, 'getting a positive example wrong' means that the system ranked it too low in the list of thirty QA candidates. For a negative example, it is just the opposite, i.e., the system ranked the example too high. The position in the rank that defines the border between *low* and *high* is the number of positive examples, according to the gold standard, in the set of thirty QA candidate pairs.

We conducted the analysis separately for the two sets below. For completeness, Table 4 (in appendix) contains the list and the classification of all positive and of all negative examples that we manually analyzed in this study.

On the positive examples

Annotation errors: Two out of the twenty examples were ranked low in the rank by all systems because they were indeed negative examples, i.e., they were mistakes by the human annotators that were not detected during the creation of the corpus. It is well known that exploring extreme examples, e.g., those wrongly classified by all systems, can help identify some of the annotation errors in corpora and, thus, it can be a good tool for corpora cleaning.

No lexical overlap: Two more examples were very challenging for the systems, as they presented no lexical overlap between the two questions and the answer, regarding content words (including derivational and morphological variations). In these examples, making the connection between the original question and the related QA pair requires semantic inference beyond simple synonymy detection, e.g., by connecting parts of the body with certain diseases or symptoms, by connecting technical terms with layperson's terms to refer to the same medical concept, etc. For the current systems to be able to handle these examples properly,

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

they would probably need more domain knowledge and more structured language representation. In principle, the embedded representations of the texts, computed by the best systems, could help bridge the lexical gap by uncovering such types of semantic relatedness from different words. We speculate that the main reasons for the failure are (i) the limited amount of data available to train the word embeddings, and (ii) the fact that the text representations were simply an aggregation of the word-based representations. Luckily, the number of examples in this category turned out to be low.

Difficult matching: In the remaining sixteen cases, one can observe some direct relations between the texts of the two questions and the answer, at the level of content words. In principle, identifying them correctly should be within reach for the current systems. However, there are a number of factors that obscure these relations. For instance, one concrete example contains broken text with several control characters. This makes it impossible for the systems to identify and to represent the words correctly. Another three examples contain unusually short answers, which are also very uninformative (e.g., 'consult your doctor' or 'answered before'). For the remaining twelve examples, a diversity of phenomena are observed to various degrees, which could explain why the systems had difficulties ranking them correctly. Among others, we can mention the lack of punctuation, pervasive spelling mistakes, words and expressions written using dialectal Arabic, unknown words, etc. All these factors affect the quality of the linguistic processing of these texts, e.g., syntactic parsing, which in turn affects the representation and the inference process to assess the semantic relatedness between the original question and the QA pair. See, for instance the following example, which contains a variety of such problems:

q_0: هل يمكن للمراة التي تعاني من متلازمة تيرنر ان تحمل بمساعدة التكنولوجيا المتقدمة الموجودة في ايامنا هذه

q_0: Is it possible for a woman who suffers from Turner Syndrome to be pregnant with the aid of advanced technology nowadays

q_s: السلام عليكم دكاترتي الاعزاء.... هل يوجد ع لا ج لمت لا زمة تيرنر وهل يوجد شي لتطويل القامة غير الابر الهرمونبة ...وهل الوزن الزائد يؤثر على عمل هذه الابر اؤجو منكم الرد وان كان يوجد نصائح العمر ١٣ الطول ١٤٣ الوزن ٦٥ .. اي نصيحة وشكرااا

q_s: Peace be upon you my dear doctors..... Is there any treatment for Tur ner Syn drome (*'Turner Syndrome' broken with some special control characters*, لمت لا زمة تيرنر) and is there anything (*dialectal writing*, شي) for lengthening tallness other than hormone needles (*spelling mistake*, الابر)... and whether extra weight affects these needles please (*spelling mistake and no punctuation*, اؤجو) reply and if there is any advice (*no punctuation*) age 13 tallness 143 weight 65—any advice and thaaanks (*repeated characters for sound effects*)

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

On the negative examples

Negative examples that were systematically ranked high by all systems were more rare. The number of positive examples in the list of thirty related QA pairs is much smaller than the number of negative cases for the development and the testing datasets, thus having all systems ranking a negative example above this number is really an extreme situation. Indeed, from the twenty selected cases, fifteen were actually annotation errors, i.e., examples that should have been annotated as *Related* instead of *Irrelevant*. The remaining five cases were truly difficult as for them there was obvious lexical and conceptual overlap between the original question and the QA pair, but the main topic of the question was not really covered in the QA pair. The systems got confused by the apparent relations at the lexical level, and failed to understand that the new question and the QA pair were about different things. For instance, in one case the original question asked about the treatment for a particular disease, and the QA pair elaborated on the difference between that disease and another one. In another question, the user asked about a treatment for a particular type of depression, while the related question is about a case of social phobia that can bring associated depression symptoms. The answer by the doctor discusses the social phobia problem, but it brings the depression into play, which might be the reason that the systems got confused. See an extended version of this example below:

q_0: (Dysthymia) تم تشخيص الاكتئاب الجزئي
عند زوجي. وهو يعالج منذ نحو شهرين وربما اكثر من ذلك بقليل بالسيبرالكس.....

q_0: The diagnostic showed that my husband has partial depression (Dysthymia), and he has had treatment for two months or slightly more using Cipralex... (*a long question*)

q_s: السؤال السلام عليكم ورحمة الله وبركاتهأعاني من الرهاب الاجتماعي
ذهبت إلى طبيب مختص في العلاج السلوكي ...كما أنه تأتيني حالات من الاكتئاب
فما هو رأيكم وهلالرهاب الاجتماعي يمكن أن لا يكون له علاج

q_s: Peace be upon you. I suffer from social phobia and I went to a specialized doctor in behavioral treatment... (*a long question*)... and sometimes I have depression so what do you think and does social phobia have a treatment

a_s: الإجابـة بسم الله الرحمن الرحيم الأخت الفاضلة ن حفظها الله.
... السلام عليكم ورحمة الله وبركاته وبعد: بالنسبة للعلاج السلوكي فهو عدة أنواع
.... السلوك الخاطئ إن كان خوفاً أو وسواساً أو حتى اكتئاباً

a_s: In the name of God... (*long greeting and introduction*): Regarding behavioral treatment, it has many types... (*a long description*)... this bad behavior if it was a fear, worry, or even depression is...

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

Discussion

The above manual analysis suggests a number of possible directions for future work aiming to improve Arabic cQA in general. First, it seems that some data cleaning is needed on the SemEval-2016 Arabic datasets as we have identified several cases of noise in the class labels.[21] Some semi-automatic procedures could be applied to identify the most likely errors and to concentrate the revision efforts on a potentially smaller dataset.

More importantly, we have several identified opportunities for improving the cQA systems in the future. Of course, they are related to the challenges described in Section 3.2 on the SemEval-2016 Arabic datasets, and they represent different levels of difficulty:

- Pre-processing and linguistic analysis should be more robust to face ungrammaticality (lack of punctuation, typos, etc.) and the use of informal language and dialectal Arabic. Reducing the number of unknown words is crucial to get meaningful representations of the texts.
- The questions and the answers in this dataset are very long (e.g., verbose greetings and introductions, long and detailed stories, etc.). The fragments that are really relevant to establish the semantic relations between texts are usually small compared to the full texts. Some focus mechanism is needed in order to concentrate on the important parts and to filter out the irrelevant. This has been the focus of research of some work on English cQA already (Lei *et al.* 2016; Barron-Cedeno *et al.* 2016; Da San Martino *et al.* 2016) and more recently on Arabic (Romeo *et al.* 2017).
- Establishing the right semantic connections between questions and answers requires the systems to improve on (i) *text representation*, in order to identify concepts linked by different expressions; and (ii) *language understanding and reasoning* at the discourse level, in order to tell whether two texts that are related at the superficial level are really discussing the same topic. The former could be at reach by using more structured domain knowledge representations (diseases, symptoms, body parts, side effects, etc.) or by training more informed specific text embeddings. The latter is much more challenging, as it is at the core of language understanding.

6.2 Contribution of syntactic information

The system results presented in Section 4.3 showed that ConvKN can improve the basic accuracy of word features by 5.52 MAP points using syntactic information (see rows 2 and 7 in Table 4.3). In this section, we analyze and provide an explanation of such remarkable impact on the reranker. For this purpose, we studied correct examples that were ranked in the top position by ConvKN when using TKs, i.e., using the $\phi_{bow}(q_o, q_s, a_s) + \phi_{tk}(q_o, q_s)$ representation and, at the same time, ranked low

[21] We have not estimated the noise level on the entire corpus; the cases that emerged from our manual analysis account for 0.23% of the examples in the dataset.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

by the ConvKN basic system, i.e., $\phi_{bow}(q_o, q_s, a_s)$, which corresponds to the ConvKN Contrastive 1 submission.

Below, we present an example in which the related question q_s is correctly ranked on top by ConvKN, but very low by ConvKN contrastive 1(rank position 21). This is the same example displayed in Figure 5

q_o: هل يوجد لاخذ الرواكيوتن لعلاج حب الشباب تاثير على الدورة الشهرية

—*Does taking Roaccutane for acne treatment have an effect on the monthly cycle?*

q_s: هل لدواء كوراكني ١٠ مغ و الخاص بعلاج حب الشباب تاثير على الدورة الشهرية

—*Does Curacne 10 mg for acne treatment have an effect on the monthly cycle?*[22]

At first glance, the example would not look complicated to be solved with standard BOW-based approaches using lexical similarities, as the two questions share several terms: *treatment*, *acne*, *effect*, *monthly*, *cycle*, etc., i.e., all non-stop words are matched except for *Roaccutane* and *Curacne*. However, the number of question/answer pairs these words occur in (a sort of document frequency) are very high: علاج (*treatment*) 5,996 occurrences, حب (*acne*) 372, الشباب (*youth*) 349, دورة (*cycle*) 4,757, شهري (*monthly*) 3,435, and تأثير (*effect*) 841, also relatively to the total number of pairs, i.e., 30,411. Therefore, according to the fundamental findings in IR regarding the low indexing capacity of highly frequent terms, they produce low similarity scores, which are not enough for the ranker to assign a higher position to such an example.

Basically, the learning algorithm will observe that such features have high score for both positive and negative examples, which would preventing it from learning useful model parameters from such examples. In contrast, a syntactic approach can intuitively measure the example similarity in terms of the syntactic structure of the two questions, thus creating precise patterns that can detect the difference between two potentially related questions. In particular, the algorithm can find specific patterns in the related questions that are distributed differently between positive and negative examples even if the terms are uniformly distributed. In the following, we explain in more detail the syntactic model used by ConvKN, analyzing the patterns it can capture to better support our hypothesis above.

Why the syntactic model works

As stated before, Figure 5 shows the syntactic trees of the original and the candidate questions discussed in the previous subsection. These parse trees are automatically obtained with the Stanford Arabic syntactic parser. The red arrows indicate a matching between words from the two questions (e.g., *Does*, *treatment*, or *effect*), whereas the blue arrows are drawn to connect entire noun phrases or clauses that are partially matched (e.g., REL-NP, REL-PP). The labels of these tree constituents

[22] Note that these are neither literal nor perfect English translations of the Arabic sentences. Still, their semantics is preserved. Also note that in the parse trees of Figure 5, we use English words closer to the literal translation. This, of course, can make the English sentence incorrect and its meaning more difficult to understand. We needed to do it to preserve the original syntactic structure of the Arabic text.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

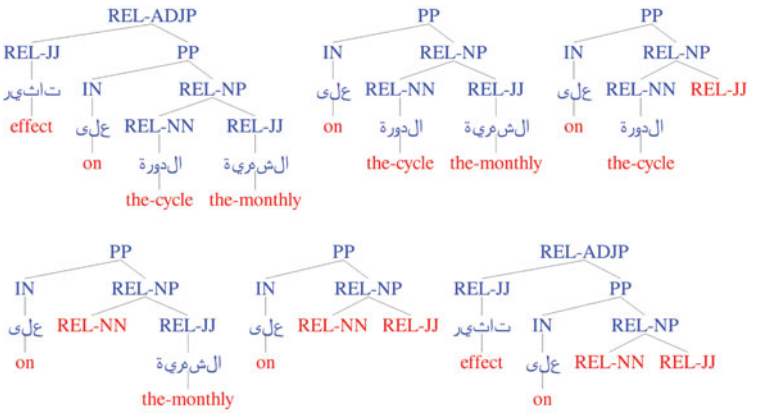


Fig. 7. (Colour online) Some patterns extracted from both parse trees of Figure 5: any pair of such fragments is a pattern for a pair of questions.

are augmented with the prefix 'REL-' to mark the connection between the syntactic trees. The two syntactic trees with the links between words and constituents form a directed acyclic graph.

Graph kernels (Vishwanathan *et al.* 2010) can measure the similarity between the graph in Figure 5 and any other graph from the training or test pairs. Thus, the kernel can use any subpart of the graph as rich, structured features, e.g., the lexico-syntactic patterns in Figure 7. Note that the structures that result from removing the lexical nodes, as well as any subtree, are more general patterns that are also used by the graph kernel as features. The more general structures help generalize the features, and provide the kernel-based learning with a back-off mechanism, from specific to more general features.

These structurally-rich features are rather useful to solve the problem of high frequency terms, which are not handled correctly by the non-syntactic version of ConvKN. For example, the following question, which is unrelated to our q_0:

q_s: السؤال السلام عليكم ورحمة الله وبركاته. سبق وأن قرأت عدة استشارات تتعلق بما يسمى الشعرانية . أعاني من نفس المشكلة لكن بشكل خفيف... لكني لا أعاني من حب الشباب والدورة الشهرية لدي منتظمة. أرجو النصيحة بخصوص أي علاج وشكرا

q_s: The question peace be upon you and God's mercy and blessings. I have read many consultations related to what's called hirsutism. I suffer from the same problem but lightly... But I don't suffer from acne and my monthly cycle is regular. Please advise for any treatment and thanks

match many terms of q_o and q_s but they do not have the same syntactic patterns and thus according to the syntactic models are different.

It should be noted that, as explained in Section 4.2, the kernel we used is not a graph kernel, it is a summation of separate TK applied to the original and candidate trees, respectively. We applied this solution because TKs are much more efficient than graph kernels, which are computationally too expensive in our problem setting. As shown in related work on question answering, the 'REL-' extended labels can

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

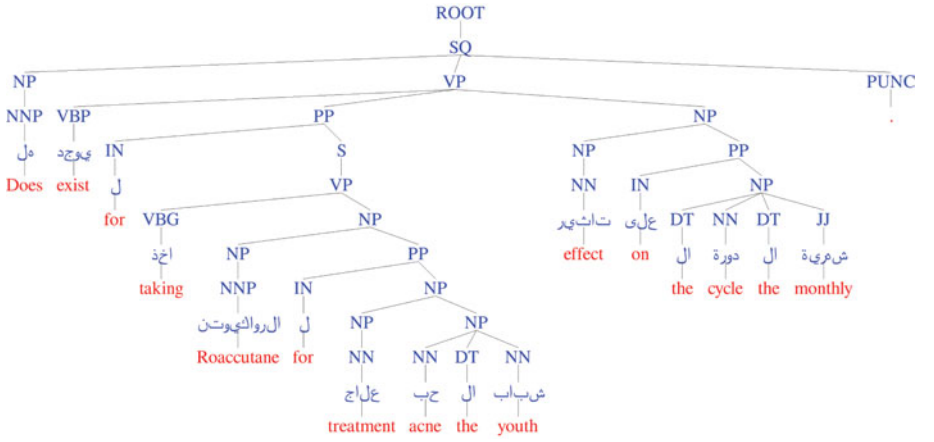


Fig. 8. (Colour online) Correct parse tree of the question q_o in Section 6.2. Compare with the incorrect upper tree in Figure 5.

work as a good surrogate of the red and blue edges connecting the two trees in the graph (Severyn and Moschitti 2012, 2013; Tymoshenko and Moschitti 2015). These labels can signal when a node is connected to another node. Although this information does not entirely replace the graph edges between trees, it has been shown to be rather effective. Additionally, TKs were shown to outperform the approximate versions of graph kernels that can be applied in practice, such as incomplete graph kernels (Filice, Da San Martino and Moschitti 2015).

Robustness of the syntactic model

One common concern arising when parsing Web text is the robustness of the parser. Clearly, non-conventional and disfluent language can impact the parser performance dramatically. Indeed, the syntactic trees shown in Figure 5 are only partially correct. For example, the syntactic tree for q_o should look like the tree depicted in Figure 8.

The main reason for the parser to fail was: (i) the informal language used by the forum users, and (ii) the flexibility of Arabic grammar, which allows free word order. In particular, the word *effect*, which in English would have been placed close to the auxiliary *does*, occurs far from it. This makes detecting the syntactic dependencies with the word *effect* very challenging. Indeed, the dependency was mistaken for both questions, q_o, and q_s.

The last observation is important as it suggests that the same types of difficult cases, probably occurring in similar sentences, tend to generate the same types of incorrect parse trees. Systematic errors can be still useful for a kernel model as they correspond to patterns that can correctly discriminate related versus unrelated questions. In the example above, *effect* is mistakenly connected with *youth acne*, leading to the wrong phrase *'youth acne effect'*, in both trees. Still, the fact that the error is repeated in both trees helps the algorithm to establish that such questions are related. In other words, the syntactic approach used with TKs, which explore all possible patterns, is also rather robust: (i) among many patterns, the learning

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

algorithm can select those that are useful for characterizing positive versus negative examples, and (ii) even if the patterns are not exactly those observed in correct parse trees, the approach is still able to use the information from partially correct parse trees, as similar errors tend to generate similar structures.

7 Conclusion

We have presented an in-depth analysis of resources and models for enabling Arabic cQA, focusing on CQA-MD, a community question answering corpus for Arabic in the domain of medical forums, and on a shared task at SemEval-2016, which used that corpus. We analyzed the systems that took part in the shared task, their features and learning algorithms, and we demonstrated with a meta-classifier that they are largely complementary, thus showing that there is large room for additional improvement. We further analyzed the corpus, which resulted in a critical discussion of what makes the task of cQA for Arabic hard in general (e.g., mixing informal and formal language, use of different Arabic dialects, etc.), and in the medical domain in particular (e.g., use of English, of complex terminology, or of layman's words to describe a complex medical term).

Moreover, we performed a careful analysis of the most frequent errors common to all systems, categorizing them into prototypical cases, and zooming into the way syntactic information in the TK approach can help solve some of the difficult cases. We believe that our analysis and the lessons learned from the process of corpus creation as well as from the shared task analysis will be helpful for future research on Arabic cQA.

In future work, we plan to verify and to clean up some potentially wrong annotations in our corpus based on our analysis above, and based on the failed predictions of multiple independent classifiers, e.g., with cross-validation. We also want to explore the retrieval of potentially related questions and answers, especially on the query formulation given the input question (Monz 2011).We further plan experiments with *Direct* and *Relevant* answers split into separate classes. Moreover, we would like to combine various information sources in the form of features directly, in a single model, as opposed to using a meta-classifier.

Moreover, given the recent success of deep neural networks, which largely eliminate the need for manual feature engineering, we are interested in exploring various deep neural network architectures, e.g., based on LSTMs, GRUs, CNNs, RNMs, etc., as well as various ways to generate high-quality word embeddings (Mihaylov and Nakov 2016b) and goodness polarity lexicons (Balchev *et al.* 2016; Mihaylov *et al.* 2017), e.g., as was done for SemEval-2016 Task 3 for English. We would also like to compare side-by-side deep neural networks to convolutional syntactic kernels, and to investigate ways to combine them. Furthermore, given the complexity of the questions, we plan to try decomposing them into simpler subquestions, similarly to (Boguraev *et al.* 2014). Another promising direction is to apply summarization techniques (Tigelaar *et al.* 2010), which could be useful both for the model and when presenting the results to the user. This might involve determining which

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

among several good answers to the target question is the most *cooperative* one, i.e., correct, useful, and non-misleading at the same time (Mendes and Coheur 2013).

Furthermore, given the abundance of data for English, we would like to try cross-language approaches (Chew *et al.* 2011; Da San Martino *et al.* 2017; Joty *et al.* 2017), combining English and Arabic data for both training and testing. Finally, given the recent proliferation of false information online, it might be worth trying to verify the factuality of the answers given in a cQA forum (Karadzhov *et al.* 2017; Nakov *et al.* 2017b; Mihaylova *et al.* 2018), as well as detecting potential forum trolls (Mihaylov *et al.* 2015a,b; Mihaylov and Nakov 2016a; Mihaylov *et al.* 2018).

References

Abdelbaki, H., Shaheen, M., and Badawy, O. 2011. ARQA high-performance arabic question answering system. In *Proceedings of the Arab Academy for Science and Technology and Maritime Transport*, ALTIC '11, Alexandria, Egypt, pp. 129–36.

Abouenour, S. 2011. On the improvement of passage retrieval in Arabic question/answering (Q/A) systems. In *Proceedings of the International Conference on Application of Natural Language to Information Systems*, Berlin, Germany, pp. 336–41.

Agichtein, E., Carmel, D., Harman, D., Pelleg, D., and Pinter, Y. 2015. Overview of the TREC 2015 LiveQA track. In *Proceedings of the Text REtrieval Conference*, TREC '15, Gaithersburg, Maryland.

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 Chalabi, H. M. 2015. *Question Processing for Arabic Question Answering System*. PhD Thesis, The British University in Dubai.

Allison, L., and Dix, T. 1986. A bit-string longest-common-subsequence algorithm. *Information Processing Letters* **23**(6): 305–10.

Balchev, D., Kiprov, Y., Koychev, I., and Nakov, P. 2016. PMI-cool at SemEval-2016 Task 3: experiments with PMI and goodness polarity lexicons for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 844–50.

Barrón-Cedeño, A., Bonadiman, D., Da San Martino, G., Joty, S., Moschitti, A., Al Obaidli, F. A., Romeo, S., Tymoshenko, K., and Uva, A. 2016. ConvKN at SemEval-2016 task 3: answer and question selection for question answering on Arabic and English fora. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 896–903.

Barrón-Cedeño, A., Filice, S., Da San Martino, G., Joty, S., Màrquez, L., Nakov, P., and Moschitti, A. 2015. Thread-level information for comment classification in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '15, Beijing, China, pp. 687–93.

Barron-Cedeno, A., Da San Martino, G., Moschitti, A., Romeo, S., Belinkov, Y., Glass, J., Mohtarami, M., Hsu, W.-N., and Zhang, Y. 2016. Neural attention for learning to rank questions in community question answering. In *Proceedings of the 26th International Conference on Computational Linguistics*, COLING '16, Osaka, Japan, pp. 1734–45.

Belinkov, Y., Mohtarami, M., Cyphers, S., and Glass, J. 2015. VectorSLU: a continuous word vector approach to answer selection in community question answering systems. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, Denver, Colorado, pp. 282–7.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

Benajiba, Y., and Rosso, P. 2007. *Arabic Question Answering*. Master's thesis, Technical University of Valencia, Spain.

Boguraev, B., Patwardhan, S., Kalyanpur, A., Chu-carroll, J., and Lally, A. 2014. Parallel and nested decomposition for factoid questions. *Natural Language Engineering*, **20**(4): 441–68.

Chew, P. A., Bader, B. W., Helmreich, S., Abdelali, A., and Verzi, S. J. 2011. An information-theoretic, vector-space-model approach to cross-language information retrieval. *Natural Lang. Eng.* **17**(1): 37–70.

Cormack, G. V., Clarke, C. L. A., and Buettcher, S. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, Boston, Massachusetts, pp. 758–9.

Cox, D. R. 1970. *The Analysis of Binary Data.* , London: Chapman and Hall.

Da San Martino, G., Romeo, S., Barrón-Cedeño, A., Joty, S., Màrquez, L., Moschitti, A., and Nakov, P. 2017. Cross-language question re-ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, Tokyo, Japan, pp. 1145–8.

Da San Martino, G., Romeo, S., Moschitti, A., and Barrón-Cedeño, A. 2016. Selecting sentences vs. selecting tree constituents for automatic question ranking. In *Proceedings of the 26th International Conference on Computational Linguistics*, COLING '16, Osaka, Japan, pp. 2515–25.

El Adlouni, Y., Lahbari, I., Rodríguez, H., Meknassi, M., and El Alaoui, S. O. 2016. UPC-USMBA at SemEval-2016 Task 3: UPC-USMBA participation in SemEval 2016 task 3, subtask D: CQA for Arabic. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 275–9.

Ezzeldin, A., 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, Amman, Jordan, pp. 1–8.

Filice, S., Croce, D., Moschitti, A., and Basili, R. 2016. KeLP at SemEval-2016 task 3: learning semantic relations between questions and answers. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 1116–23.

Filice, S., Da San Martino, G., and Moschitti, A. 2015. Structural representations for learning relations between pairs of texts. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '15, Beijing, China, pp. 1003–13.

Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, **29**(5): 1189–232.

Green, S., and Manning, C. D. 2010. Better Arabic parsing: baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, Beijing, China, pp. 394–402.

Guzmán, F., Joty, S., Màrquez, L., and Nakov, P. 2015. Pairwise neural machine translation evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '15, Beijing, China, pp. 805–14.

Guzmán, F., Màrquez, L., and Nakov, P. 2016a. Machine translation evaluation meets community question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, Berlin, Germany, pp. 460–6.

Guzmán, F., Nakov, P., and Màrquez, L. 2016b. MTE-NN at SemEval-2016 Task 3: can machine translation evaluation help community question answering? In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 887–95.

Hammo, B., Abu-Salem, H., and Lytinen, S. 2002. QARAB: a question answering system to support the Arabic language. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, SEMITIC '02, Philadelphia, Pennsylvania, pp. 1–11.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

Hoogeveen, D., Verspoor, K. M., and Baldwin, T. 2015. CQADupStack: a benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian Document Computing Symposium*, ADCS '15, Sydney, Australia, pp. 1–8.

Hoque, E., Joty, S., Màrquez, L., Barrón-Cedeño, A., Da San Martino, G., Moschitti, A., Nakov, P., Romeo, S., and Carenini, G. 2016. An interactive system for exploring community question answering forums. In *Proceedings of the 26th International Conference on Computational Linguistics*, COLING '16, Osaka, Japan, pp. 1–5.

Jaccard, P. 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles* **37**: 547–79.

Joachims, T. 1999. Making large-scale support vector machine learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, (eds.), *Advances in Kernel Methods*, pp. 169–84. MIT Press.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, Edmonton, Canada, pp. 133–42.

Joty, S., Barrón-Cedeño, A., Da San Martino, G., Filice, S., Màrquez, L., Moschitti, A., and Nakov, P. 2015. Global thread-level inference for comment classification in community question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP '16, Lisbon, Portugal, pp. 573–8.

Joty, S., Màrquez, L., and Nakov, P. 2016. Joint learning with global inference for comment classification in community question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '16, San Diego, California, pp. 703–13.

Joty, S., Nakov, P., Màrquez, L., and Jaradat, I. 2017. Cross-language learning with adversarial neural networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, CoNLL '17, Vancouver, Canada, pp. 226–37.

Kanaan, G., and Hammouri, A. 2009. A new question answering system for the Arabic language. In *Proceedings of the American Journal of Applied Sciences* **6**(4): 797–805.

Karadzhov, G., Nakov, P., Màrquez, L., Barrón-Cedeño, A., and Koychev, I. 2017. Fully automated fact checking using external sources. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP '17, Varna, Bulgaria, pp. 344–53.

Lei, T., Joshi, H., Barzilay, R., Jaakkola, T., Tymoshenko, K., Moschitti, A., and Màrquez, L. 2016. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '16, San Diego, California, pp. 1279–89.

Magooda, A., Gomaa, A., Mahgoub, A., Ahmed, H., Rashwan, M., Raafat, H., Kamal, E., and Al Sallab, A. 2016. RDI at SemEval-2016 Task 3: RDI unsupervised framework for text ranking. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 822–7.

Malhas, R.,Torki, M., and Elsayed, T. 2016. QU-IR at SemEval-2016 Task 3: learning to rank on Arabic community question answering forums with word embedding. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 866–71.

Mendes, A. C., and Coheur, L. 2013. When the answer comes into question in question-answering: survey and open issues. *Natural Language Engineering* **19**(1): 1–32.

Mihalcea, R., and Tarau, P. 2004. TextRank: bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '04, Barcelona, Spain, pp. 404–11.

Mihaylov, T., Balchev, D., Kiprov, Y., Koychev, I., and Nakov, P. 2017. Large-scale goodness polarity lexicons for community question answering. In *Proceedings of the 40th International*

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, Tokyo, Japan, pp. 1185–8.

Mihaylov, T., Georgiev, G., and Nakov, P. 2015a. Finding opinion manipulation trolls in news community forums. In *Proceedings of the 19th Conference on Computational Natural Language Learning*, CoNLL '15, Beijing, China, pp. 310–4.

Mihaylov, T., Koychev, I., Georgiev, G., and Nakov, P. 2015b. Exposing paid opinion manipulation trolls. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, RANLP '15, Hissar, Bulgaria, pp. 443–50.

Mihaylov, T., Mihaylova, T., Nakov, P., Màrquez, L., Georgiev, G., and Koychev, I. 2018. The dark side of news community forums: opinion manipulation trolls. *Internet Research* **28**(5): 1292–1312.

Mihaylov, T., and Nakov, P. 2016a. Hunting for troll comments in news community forums. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, Berlin, Germany, pp. 399–405.

Mihaylov, T., and Nakov, P. 2016b. SemanticZ at SemEval-2016 Task 3: ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 879–86.

Mihaylova, T., Gencheva, P., Boyanov, M., Yovcheva, I., Mihaylov, T., Hardalov, M., Kiprov, Y., Balchev, D., Koychev, I., Nakov, P., Nikolova, I., and Angelova, G. 2016. SUper team at SemEval-2016 task 3: building a feature-rich system for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 836–43.

Mihaylova, T., Nakov, P., Màrquez, L., Barrón-Cedeño, A., Mohtarami, M., Karadjov, G., and Glass, J. 2018. Fact checking in community forums. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, AAAI '18, New Orleans, Louisiana, pp. 879–86.

Mikolov, T., Kombrink, S., Burget, L., Černocký, J., and Khudanpur, S. 2011. Extensions of recurrent neural network language model. In *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '11, Prague, Czech Republic, pp. 5528–31.

Mikolov, T., Yih, W.-T., and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '13, Atlanta, Georgia, pp. 746–51.

Mohtarami, M., Belinkov, Y., Hsu, W.-N., Zhang, Y., Lei, T., Bar, K., Cyphers, S., and Glass, J. 2016. SLS at SemEval-2016 task 3: neural-based approaches for ranking in community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 828–35.

Monz, C. 2011. Machine learning for query formulation in question answering. *Natural Language Engineering* **17**(4): 425–54.

Moschitti, A. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th European Conference on Machine Learning*, ECML '06, Berlin, Germany, pp. 318–29.

Moschitti, A., Màrquez, L., Nakov, P., Agichtein, E., Clarke, C., and Szpektor, I. 2016. SIGIR 2016 Workshop WebQA II: web question answering beyond factoids. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, Pisa, Italy, pp. 1251–2.

Nakov, P., Hoogeveen, D., Màrquez, L., Moschitti, A., Mubarak, H., Baldwin, T., and Verspoor, K. 2017a. SemEval-2017 task 3: community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, SemEval '17, Vancouver, Canada, pp. 27–48.

Nakov, P., Màrquez, L., and Guzmán, F. 2016a. It takes three to tango: triangulation approach to answer ranking in community question answering. In *Proceedings of the 2016*

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

Conference on Empirical Methods in Natural Language Processing, EMNLP '16, Austin, Texas, pp. 1586–97.

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. A., Glass, J., and Randeree, B. 2016b. SemEval-2016 task 3: community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 525–45.

Nakov, P., Mihaylova, T., Màrquez, L., Shiroya, Y., and Koychev, I. 2017b. Do not trust the trolls: predicting credibility in community question answering forums. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP '17, Varna, Bulgaria, pp. 551–60.

Nicosia, M., Filice, S., Barrón-Cedeño, A., Saleh, I., Mubarak, H., Gao, W., Nakov, P., Da San Martino, G., Moschitti, A., Darwish, K., Màrquez, L., Joty, S., and Magdy, W. 2015. QCRI: answer selection for community question answering – experiments for Arabic and English. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, Denver, Colorado, pp. 203–9.

Qiu, X., and Huang, X. 2015. Convolutional neural tensor network architecture for community-based question answering. In *Proceedings of International Joint Conference on Artificial Intelligence*, IJCAI '15, Buenos Aires, Argentina, pp. 1305–11.

Ray. S. K., and Shaalan, K. 2016. A review and future perspectives of Arabic question answering systems. *IEEE Transactions on Knowledge and Data Engineering* **28**(12): 3169–90.

Romeo, S., Da San Martino, G., Belinkov, Y., Barrn-Cedeo, A., Eldesouki, M., Darwish, K., Mubarak, H., Glass, J., and Moschitti, A. 2017. Language processing and learning models for community question answering in Arabic. *Information Processing & Management*. `https://www.sciencedirect.com/science/article/abs/pii/S0306457316306720`

Rosso, P., Lyhyaoui, A., Peñarrubia, J., Montes y Gómez, M., Benajiba, Y., and Raissouni, N. 2005. Arabic-English question answering. In *Proceedings of the Information Communication Technologies International Symposium*, ICTIS '05, Tetuan, Morocco.

Salem, Z., Sadek, J., Chakkour, F., and Haskkour, N. 2010. Automatically finding answers to "why" and "how to" questions for Arabic language. In *Proceedings of the International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, KES '10, Cardiff, United Kingdom, pp. 586–93.

dos Santos, C., Barbosa, L., Bogdanova, D., and Zadrozny, B. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '15, Beijing, China, pp. 694–9.

Severyn, A., and Moschitti, A. 2012. Structural relationships for large-scale learning of answer re-ranking. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, Portland, Oregon, pp. 741–50.

Severyn, A., and Moschitti, A. 2013. Automatic feature engineering for answer selection and extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, Seattle, Washington, pp. 458–67.

Severyn, A., and Moschitti, A. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, Santiago, Chile, pp. 373–82.

Severyn, A., Moschitti, A., Tsagkias, M., Berendsen, R., and de Rijke, M. 2014. A syntax-aware re-ranker for microblog retrieval. In *Proceedings of the 37th International ACM*

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast, Australia, pp. 1067–70.

Shen, Y., Rong, W., Sun, Z., Ouyang, Y., and Xiong, Z. 2015. Question/answer matching for CQA system via combining lexical and sequential information. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, AAAI '15, Austin, Texas, pp. 275–81.

Silberztein, M. 2005. NOOJ: a linguistic annotation system for corpus processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, HLT/EMNLP '05, Vancouver, Canada, pp. 10–1.

Tigelaar, A. S., Op Den Akker, R., and Hiemstra, D. 2010. Automatic summarisation of discussion fora. *Natural Language Engineering* **16**(2): 161–92.

Trigui, O., Belguith, H. L., and Rosso, P. 2010. DefArabicQA: arabic definition question answering system. In *Proceedings of the 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 on Conference on Information and Knowledge Management*, CIKM '15, Melbourne, Australia, pp. 1451–60.

Vishwanathan, S. V. N., Schraudolph, N. N., Kondor, R., and Borgwardt, K. M. 2010. Graph kernels. *Journal of Machine Learning Research* **11**: 1201–42.

Wang, D., and Nyberg, E. 2015. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '15, Beijing, China, pp. 707–12.

Wise, M. 1996. YAP3: improved detection of similarities in computer program and other texts. In *Proceedings of the 27th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '96, New York, NY, pp. 130–4.

Wissal, B., Ellouze, M., Mesfar, S., and Belguith, L. H. 2009. An Arabic question-answering system for factoid questions. In *Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering*, NLP-KE '09, Dalian, China, pp. 1–7.

Wu, Y., and Zhang, M. 2016. ICL00 at SemEval-2016 task 3: translation-based method for CQA system. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, pp. 857–60.

Zhou, G., He, T., Zhao, J., and Hu, P. 2015. Learning continuous word embedding with metadata for question retrieval in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '15, Beijing, China, pp. 250–9.

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press

Table 4. *IDs of the most difficult examples that were manually analyzed. They are grouped according to the discussion in Section 6*

Positive Examples		
Annotation errors (2)		200553-68316, 201053-29515
No lexical match (2)		201566-101125, 200100-20458
Difficult matching	Broken (1)	200407-35856
	Short (3)	201430-28966, 200975-35649, 200553-7105
	Other (12)	200975-22184, 201324-52971, 201375-47830, 200789-50972, 200374-39477, 201413-26488, 200629-15532, 201524-37828, 200185-10569, 200058-31957, 200763-47614, 200549-34213
Negative Examples		
Annotation errors (15)		200728-306863, 200471-329176, 200792-307304, 200471-317305 201525-315764, 200415-675932, 00974-8326, 200628-309699 200471-35743, 200655-320798, 201458-321659, 201399-322993 201422-324147, 201085-319095
Misleading matching (5)		00974-325572, 201285-49484, 201517-310016 201093-300076, 200412-329233

Appendix

Table 4 shows the IDs of the most difficult examples that we manually analyzed (cf. Section 6). They are split into the Positive set (low-ranked 'D' or 'R' examples) and Negative set (high-ranked 'I' examples). The IDs contain the original question ID and the related question-answer pair, separated by a dash symbol. They are grouped according to the discussion in Section 6. Within each group, they are sorted in decreasing order of difficulty. The complete examples can be downloaded from the SemEval-2016 Task 3 website.[23]

[23] `http://alt.qcri.org/semeval2016/task3/index.php?id=data-and-tools`

https://doi.org/10.1017/S1351324918000426 Published online by Cambridge University Press