UPC-USMBA at SemEval-2017 Task 3: Combining Multiple Approaches for CQA for Arabic

Yassine El Adlouni
USMBA
Fes, Morocco
yeladlouni@gmail.com

Imane Lahbari
USMBA
Fes, Morocco
imane.lahbari@usmba.ac.ma

Horacio Rodríguez
UPC
Barcelona, Spain
horacio@cs.upc.edu

Mohammed Meknassi
USMBA
Fes, Morocco
m.meknassi@gmail.com

Said Ouatik El Alaoui
USMBA
Fes, Morocco
s_ouatik@yahoo.com

Noureddine Ennahnahi
USMBA
Fes, Morocco
nahnnourd@yahoo.fr

Abstract
This paper presents a description of the participation of the UPC-USMBA team in the SemEval 2017 Task 3, subtask D, Arabic. Our approach for facing the task is based on a performance of a set of atomic classifiers (lexical string-based, vectorial, and rule-based) whose results are later combined. Our primary submission has obtained good results: 2nd (from 3 participants) in MAP, and 1st in in accuracy.

1 Introduction
The SemEval Task 3 subtask D, (Nakov et al., 2017), asks, given a query, consisting of a question, and a set of 30 question-answer pairs, to re-rank the question-answer pairs according to their relevance with respect to the original question.

Question Answering, QA, i.e. querying a computer using Natural Language, is a traditional objective of Natural Language Processing. CQA differs from conventional QA systems basically on three aspects: The source of the possible answers, that are the threads of queries and answers activated from the original query, the structure of the threads and the available metadata can be exploited for the task, types of questions include the frequent use of complex questions, as definitional, why, consequences, how_to_proceed, etc. One factor that makes very attractive the task is that many approaches, rule-based, pattern-based, Statistical, ML, have been applied to face it. See (Nakov et al., 2017) for an overview of frequently used techniques. See also the overviews of past contests, (Nakov et al., 2016a) and (Nakov et al., 2016b).

2 Our Approach
Due to the negative results in last year participation, for this year we present a system that combines different classifiers, going beyond the two classifiers, Arabic and English shallow features-based ones, used last year. The new classifiers follow approaches that have produced good results in systems as (Barrón-Cedeño et al., 2016), (Mihaylov and Nakov, 2016), and (Joty et al., 2016). We will refer in what follows to these classifiers as atomic ones and they are further combined for obtaining the final results.
The overall architecture of our system is presented in Figure 1. As can be seen, the system performs in four steps, a preliminary step, aiming at collecting needed resources, basically Arabic and English classified medical terminologies, a learning step, for getting the models, a classification step, for applying them to the test dataset, and a last step combining the results of the atomic classifiers that are described next.

2.1 Overall description

A core component of our approach is the use of a medical terminology, covering both Arabic and English terms and organized into three categories: body parts, drugs, and diseases. See (Adlouni et al., 2016).

After downloading the training (resp. test) Arabic dataset we translate into English all the Arabic query texts and all the Arabic texts corresponding to each of the query/answers pairs. For doing so we have used the Google Translate API. The texts are then processed using for English the Stanford CoreNLP toolbox (Manning et al., 2014) and for Arabic Madamira (Pasha et al., 2015). The results are then enriched with WordNet synsets and with Named Entities included in the medical dictionaries for both Arabic and English. Then a process of feature extraction is carried out. This process is different for each atomic classifier and will be described next. Finally, a process of learning (resp. classification) is performed. Also these processes differ depending on the involved classifiers.

2.2 Atomic Classifiers

The atomic classifiers used by our system are the following:

- **Basic lexical string-based classifiers**, i.e. Basic_ar and Basic_en, identical to the ones used last year. The basic classifiers use three sets of features: shallow linguistic features, vectorial features, and domain-based features. Details can be seen in (Adlouni et al., 2016). We have used for learning the

Logistic Regression classifier included in the Weka toolkit, (Hall et al., 2009).

- A simple IR system, using LUCENE engine, with different combinations served as index, Question, Answer and Question concatenated with the Answer.

- Latent Semantic Indexing (LSI), learned from different datasets, was used to get dense representations of our sentences by using SVD (Singular Value Decomposition). These vectorial representations are then used to measure the similarity between each pair \(Q_o/Q_i\) where \(Q_o\) denote the original question and \(Q_i\) denote the \(i^{th}\) Question within the set of questions to rank. Various corpora was used for that matter including Wikipedia, Webteb.com, alitibbi.com and dailymedicalinfo.com which are specialized Arabic websites for medical domain articles. The pre-processing step consisted of denoising collected articles, extracting paragraphs, removing stopwords, diacritics, tokenizing, normalizing and lemmatizing. The same pipeline is used later for the query and for each pair of Question/Answer. The implementation used for LSI is from gensim (Rehurek and Sojka, 2010). After the SVD decomposition, cosine similarity measure is calculated for each pair which are ordered for each query and a quartile approach is taken to decide if the pair is relevant or not.

- A topic-based LDA using the same training datasets that for LSI. We used the implementation of Rehurek’s gensim.

- Embedding systems. We have tried several embeddings with no remarkable results. Specifically we tried Word2Vec, Glove, and doc2vec. The last one produced the best results but was outperformed by the combination of LDA and LSI.

- A Rulebased system, with rulesets for Arabic and English. The motivation of rule-based classifiers is that for some queries both the original questions and some of the questions that for LSI. We used the implementation of Rehurek’s gensim.

A simple IR system, using LUCENE engine, with different combinations served as index, Question, Answer and Question concatenated with the Answer.

- Latent Semantic Indexing (LSI), learned from different datasets, was used to get dense representations of our sentences by using SVD (Singular Value Decomposition). These vectorial representations are then used to measure the similarity between each pair \(Q_o/Q_i\) where \(Q_o\) denote the original question and \(Q_i\) denote the \(i^{th}\) Question within the set of questions to rank. Various corpora was used for that matter including Wikipedia, Webteb.com, alitibbi.com and dailymedicalinfo.com which are specialized Arabic websites for medical domain articles. The pre-processing step consisted of denoising collected articles, extracting paragraphs, removing stopwords, diacritics, tokenizing, normalizing and lemmatizing. The same pipeline is used later for the query and for each pair of Question/Answer. The implementation used for LSI is from gensim (Rehurek and Sojka, 2010). After the SVD decomposition, cosine similarity measure is calculated for each pair which are ordered for each query and a quartile approach is taken to decide if the pair is relevant or not.

- A topic-based LDA using the same training datasets that for LSI. We used the implementation of Rehurek’s gensim.

- Embedding systems. We have tried several embeddings with no remarkable results. Specifically we tried Word2Vec, Glove, and doc2vec. The last one produced the best results but was outperformed by the combination of LDA and LSI.

- A Rulebased system, with rulesets for Arabic and English. The motivation of rule-based classifiers is that for some queries both the original questions and some of the questions...
included into the thread are short questions involving a clear objective. We can manually build condition rules for recognizing these questions and extracting their objectives. Consider, for instance, a question beginning with “What is the cause of”, and containing close to it a disease name. This question can be easily classified with the Question type \(QT\) CauseDisease and parameterized with the tag Disease with the extracted name as value. Similarly we can build answer rules for detecting whether the answer part of a pair satisfy the objective (in this example) an occurrence of the disease name. If the original question fires a condition rule and is classified with a \(QT\) with some associated tag and some of its questions within the thread are also classified with the same \(QT\) being their tags compatible, it is highly likely that the corresponding pairs are relevant for the original query. Moreover, if the answer part of the pair satisfy the associated answer rule the confidence (and, thus, the score) of the pair increases. Unfortunately although the precision of condition rules is high, recall is very low. Our hope is that with careful engineering of the rules and this kind of atomic classifier if not alone could contribute to improve the performance of other classifiers. 13 QTs were used for Arabic and 16 QTs for English, with a total of 75 rules.

### 2.3 Combinations

Output of the atomic classifiers are further combined. We have evaluated the powerset of the atomic classifiers for looking for the best combination using the training set. However, no more than 3 atomic classifiers produced good results and the best one resulted from the combination of one of the LSI and one of the LDA classifiers. The parameters used for learning the combiner are the following:

- **scoring form**, i.e. 'max' or 'ave', defining how for each pair \(i\) of each query \(q\) the scores of the different atomic components \(s\) are combined.
- **thresholding form**, i.e. None, 'global' or 'local', defining whether a threshold has to be used for getting the result of each pair \(i\).
- **thresholding level**, i.e. 0.2, 0.4, 0.6, 0.8.
- **result form**, i.e. 'max', 'voting', 'coincidence'.
combinations of atomic classifiers. The best results were obtained for the combination LDA and LSI learned from Webteb, lemmatized. This combination was our primary run. As we were interested on the performance of our manual rules we submitted, too, a contrastive run including a combination of basic_ar and basic_en with rule_based.

We were interested on analyzing two measures MAP as official measure and accuracy as the measure based on the individual results and not in the order. As our classifiers are not true rankers, analyzing the two measures seemed more appropriate for evaluating our system and proposing ways of improvement.

4 Results

In Table 1 a summary of the Official results of Semeval 2017 Task 3 Subtask D, corresponding to primary runs is presented.

Regarding MAP, and so, looking at the official rank, we are placed in the middle (2nd from 3 participants). Regarding accuracy, that is important for us as argued in previous section, we are placed on the top of the rank. We analyzed the results in the test dataset of our atomic classifiers (with different parameterization) and combinations. Due to space constraints we cannot include the whole results. The MAP for the atomic classifiers (using the best parameters got in training) range from 55 to 58.32. All the atomic results were outperformed by our primary run but Lucene obtaining our best result, 58.32.

5 Conclusions and future work

This year our results have been rather good, second (but from only 3 teams) in MAP and first in accuracy.

From our contrastive run we need more time for analyzing the results. The accuracy of each rule of each language should be measured and some rules should be refined, some others removed and probably more rules are needed.

Our next steps will be:

• Performing an in depth analysis of the performance of our two rulesets, analyzing the accuracy of each rule and cross comparing the rules fired in each language. It is likely that if a rule has been correctly applied to a pair for a language a corresponding rule in the other languages should be applied as well, so modifying an existing rule or including a new one could be possible. Learning a rule classifier is another possibility to examine.

• Using a final ranker over the results of our atomic classifiers for trying to improve our MAP.

• Trying others NN models as CNN and LSTM.

• Extending the coverage of our medical terminologies to other medical entities (procedures, clinical signs, etc).

Acknowledgments

We are grateful for the comments and suggestions from four anonimous reviewers. Dr. Rodríguez has been partially funded by Spanish project "GraphMed" (TIN2016-77820-C3-3R).
References


