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 rerank 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.

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 275–279,
Vancouver, Canada, August 3 - 4, 2017.

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[1]. The texts are then processed using for English the Stanford CoreNLP toolbox[2] (Manning et al., 2014) and for Arabic *Madamira*[3] (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[4] 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[5]: 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[6], (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^{st} Question within the set of questions to rank. Various corpora was used for that matter including Wikipedia, Webteb.com, altibbi.com and dailymedicalinfo.com which are specialized Arabic websites for medical domain articles. The preprocessing 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 (Řehůřek 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[7].

- *Embedding* systems. We have tried several embeddings with no remarkable results. Specifically we tried *Word2Vec*[8], *Glove*[9], and *doc2vec*[10]. 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

[1] translate.google.com

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

[3] http://nlp.ldeo.columbia.edu/madamira/

[4] In fact the classifiers, besides classifying each pair as relevant or not, use their confidence scores for obtaining the score of each pair and, thus, their relative order. We can, so, define them as regressors or rankers.

[5] Extracted independently for each language.

[6] http://www.cs.waikato.ac.nz/ml/weka/

[7] http://radimrehurek.com/gensim/models/ldamodel.html

[8] http://deeplearning4j.org/word2vec.html

[9] http://nlp.stanford.edu/projects/glove/

[10] http://radimrehurek.com/gensim/models/doc2vec.html

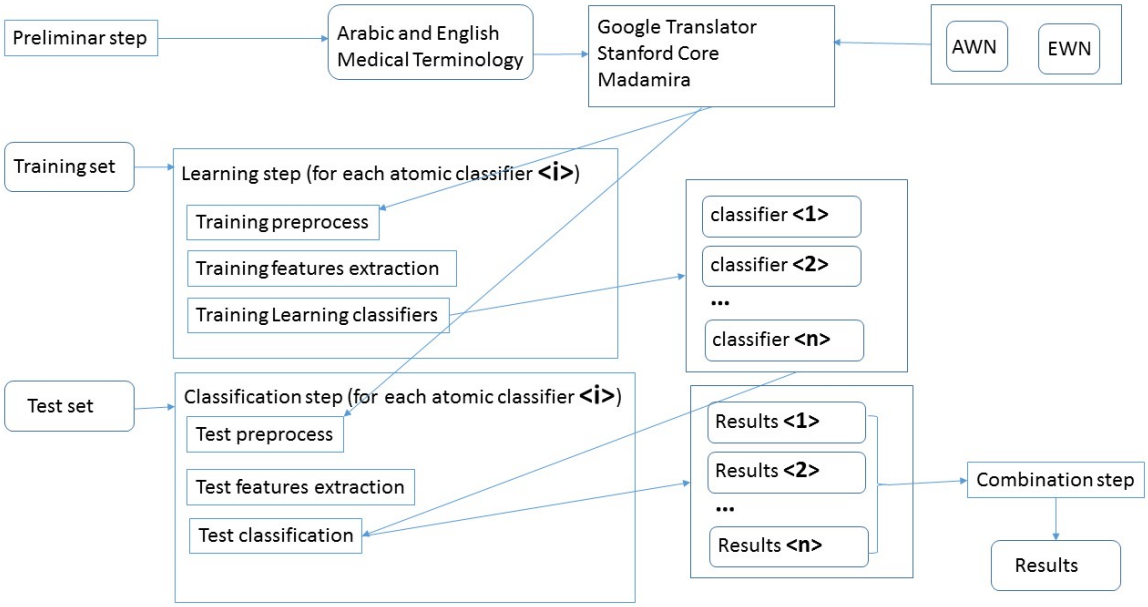

Figure 1: Train and testing pipelines

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'.

3 Experimental framework

We carried out all the processes depicted in Figure 1, for preprocessing and training using the training dataset. Besides, we tried all the possible

Team	Rank	MAP	Accuracy
GW_QA-primary	1	0.6116	0.6077
UPC-USMBA-primary	2	0.5773	0.6624
QU_BIGIR-primary	3	0.56695	0.4964

Table 1: Official results of the task

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 argumented 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

Yassine El Adlouni, Imane Lahbari, Horacio Rodríguez, Mohammed Meknassi, Said Ouatik El Alaoui, and Noureddine Ennahnahi. 2016. Upc-usmba participation in semeval 2016 task 3, subtask d: Cqa for arabic. In *NAACL HLT 2016, At San Diego, CA, Volume: In Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval 16*.

Alberto Barrón-Cedeño, Giovanni Da San Martino, Shafiq R. Joty, Alessandro Moschitti, Fahad Al-Obaidli, Salvatore Romeo, Kateryna Tymoshenko, and Antonio Uva. 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@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 896–903.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian Witten. 2009. The WEKA Data Mining Software: An Update. In *SIGKDD Explorations*.

Shafiq R. Joty, Lluís Màrquez, and Preslav Nakov. 2016. Joint learning with global inference for comment classification in community question answering. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 703–713.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Todor Mihaylov and Preslav Nakov. 2016. 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@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 879–886.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016a. SemEval-2016 Task 3: Community Question Answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June. Association for Computational Linguistics.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016b. Semeval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 525–545.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, SemEval '17, Vancouver, Canada, August. Association for Computational Linguistics.

Arfath Pasha, Mohammad Al-Badrashiny, Mona Diab, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2015. Madamira 2.1. In *Center for Computational Learning Systems Columbia University, April 2015*, pages 55–60.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA.