QU-BIGIR at SemEval 2017 Task 3: Using Similarity Features for Arabic Community Question Answering Forums

Marwan Torki and **Maram Hasanain** and **Tamer Elsayed**
Qatar University
Department of Computer Science & Engineering
Doha, Qatar
{mtorki,maram.hasanain,telsayed}@qu.edu.qa

Abstract

In this paper, we describe our QU-BIGIR system for the Arabic subtask D of the SemEval 2017 Task 3. Our approach builds on our participation in the past version of the same subtask. This year, our system uses different similarity features that encodes lexical and semantic pairwise similarity of text pairs. In addition to well-known similarity measures such as cosine similarity, we use other measures based on the summary statistics of word embedding representation for a given text. To rank a list of candidate question-answer pairs for a given question, we train a linear SVM classifier over our similarity features. Our best resulting run came second in subtask D with a very competitive performance to the first-ranking system.

1 Introduction

The ubiquitous presence of community question answering (CQA) websites has motivated research on building automatic question answering (QA) systems that can benefit from previously-answered questions to answer newly-posed ones (Shtok et al., 2012). A core functionality of such systems is their ability to effectively rank previously-suggested answers with respect to their degree/probability of relevance to a posted question. Ranking is vital to push away irrelevant and low quality answers, which are commonplace in CQA as they are generally open with no restrictions on who can post or answer questions.

To this effect, SemEval 2017 Task 3 "Community Question Answering" has emphasized the ranking component in the main task of the challenge. We have participated in Task 3-Subtask D (Arabic Subtask) which is confined to the main task of ranking answers; given a new question and a set of 30 question-answer pairs (QApairs) retrieved by a search engine, re-rank those QApairs by their degree/probability of relevance to the new question. Figure 1 shows an example of a question and four of its 30 given candidate question-answer pairs. Further details about SemEval 2017 Task 3 can be found in (Nakov et al., 2017).

Question: ما هي الدوالي وما هي اسبابها

Candidate question-answer pairs (QApairs):

Q : هل لدوالي القدم اعراض يوجد معي دوالي في القدم ولكن اطلب منكم تزويدي باعراض الدوالي والمشاكل التي يسببها
A : العرض الأكثر شيوعا هو ظهور الأوردة منتفخة على واجهة الساق والتشخيص يكون من خلال الفحص الطبي المباشر والعلاج سيكون وفق السبب

Q : ماهى دوالى الدرجه الثالثه وما اسبابها وكيف يتم الوقايه منها
A : الدوالي هي عباره عن توسع في الاورده العلويه للمنطقه المصابه ولها الكثير من الاسباب كالحمل والسمنه والجلوس لمده طويله جدا ويتم الوقايه منها عن طريق عدم الوقوف لمده طويله وممارسة الرايضه بشكل اكبر وليس جوارب مخصصه لهذه الحالات لمنع التفاقم

Q: هل العروق الملتويه بشكل واضح في القدم تعتبر من الدوالي وهل الدوالي لها اخطار اذا لم يتم علاجها
A: يمكن أن تنفجر الدوالي تحت الجلد وتؤدى إلى كدمات وترسبات دموية وتصبغات تحت الجلد ويمكن أن تؤدي إلى تقرحات. قد يشمل علاج دوالي الساقين عدد من التدابير التي تهدف الى التخفيف من حدة الأعراض أو الحد من تطورها

Q: انا شاب عمري ٣٧سنة وأعاني من من دوالي الأوردة في الساقين فكيف أزيلها
A: الغالب أنه بحاجة لاستئصال إذا كانت كبيرة ومزعجة راجع اختصاصي جراحة عامة أو جراحة الأوعية الدموية لتقييم الحالة

Figure 1: A question and 4 of its given 30 candidate QA-pairs

In this paper, we describe the system we developed to participate in that task. The system leverages a supervised learning approach over similarity features. We utilize two types of similarity features. First, we employ similarity features based on term representation for a given pairs of text. Second, we utilize word2vec to build text representation following the same approach as in our last year's submission for the same subtask (Malhas et al., 2016). We used similarity features based on that text representation to encode the semantic similarity for pairs of texts.

The rest of the paper is organized as follows; the approach and description of features are in-

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 360–364,
Vancouver, Canada, August 3 - 4, 2017.

troduced in section 2; the experimental setup followed in our submitted runs and the results are presented in section 3. Finally we conclude our study with final remarks in section 4.

2 Approach

We tackled the answer ranking task with a supervised learning approach that uses linear SVM models. The features used in classification are designed to capture both lexical and semantic information of pairs of texts.

2.1 Data Setup

We are given a set of questions Q; each is associated with P question-answer pairs. To compute our features, we define a text pair $< T_1, T_2 >$ according to three setups:

- **QQA:** We consider T_1 to be the original question q and the concatenation of one **pair** p of its associated question-answer pairs as T_2.
- **QA:** We consider T_1 to be the original question q and one **answer** of its associated question-answer pairs as T_2.
- **QQ:** We consider T_1 to be the original question q and one **question** of its associated question-answer pairs as T_2.

2.2 Term-based Similarity Features

A recent study has showed that simple features like MK features (Metzler and Kanungo, 2008) can be very effective for re-ranking candidate question answer pairs (Yang et al., 2016). We specifically use the following features described by Yang et al. (Yang et al., 2016). For all features, we assume the input to be two pieces of text: T_1 and T_2 as defined by any of the setups illustrated in section 2.1.

- **SynonymsOverlap:** Before computing this feature, we first apply light text normalization to T_1 and T_2 including special character (e.g., ',', '.', etc.) and diacritics removal. The feature is then computed as the portion of T_1 terms that have a synonym or the original term in T_2. Synonyms are extracted from the Arabic WordNet.[1]

To compute the remaining features, we normalize T_1 and T_2 following the same approach when computing SynonymsOverlap feature. We also apply preprocessing steps including stemming and stopwords removal.

- **LMScore:** The language model score is computed as the Dirichlet-smoothed log-likelihood score of generating T_1 given T_2. The score is computed using the following equation:

$$LMScore(T_1, T_2) = \sum_{w \in T_1} tf_{w,T_1} \log \frac{tf_{w,T_2} + \mu P(w|C)}{|T_2| + \mu} \quad (1)$$

where tf_{w,T_1} and tf_{w,T_2} is the frequency of term w in T_1 and T_2 respectively. $P(w|C)$ is the background language model computed using the maximum likelihood estimate with term statistics extracted from a recent large-scale crawl of the Arabic Web called ArabicWeb16 (Suwaileh et al., 2016). We set μ to 2000 as this is the default value used in Lucene's language modeling retrieval model.[2]

- **CosineSimialirty**. This feature computes the cosine similarity between T_1 and T_2 as follows.

$$CS(T_1, T_2) = \frac{\vec{T_1} \cdot \vec{T_2}}{||\vec{T_1}|| \, ||\vec{T_2}||} \quad (2)$$

where $\vec{T_1}$ and $\vec{T_2}$ is the vector representation of T_1 and T_2 respectively and $||\vec{T_1}||$ and $||\vec{T_2}||$ is the Euclidean lengths of vectors $\vec{T_1}$ and $\vec{T_2}$. We represent texts as vectors using TF-IDF representation where term statistics are extracted from ArabicWeb16 (Suwaileh et al., 2016).

- **JaccardSimialirty**. This feature computes the Jaccard similarity between T_1 and T_2 as follows.

$$JS(T_1, T_2) = \frac{|\vec{T_1} \cap \vec{T_2}|}{|\vec{T_1} \cup \vec{T_2}|} \quad (3)$$

- **JaccardSimialirtyV1**. This is a variant of Jaccard similarity computed as follows.

$$JS_1(T_1, T_2) = \frac{|\vec{T_1} \cap \vec{T_2}|}{|\vec{T_1}|} \quad (4)$$

[1] Described here: http://bit.ly/2mzfc7X

[2] http://bit.ly/2lOOdqw

- **JaccardSimialirtyV2**. This is a second variant of Jaccard similarity computed as follows.

$$JS_2(T_1, T_2) = \frac{|\vec{T_1} \cap \vec{T_2}|}{|\vec{T_2}|} \quad (5)$$

2.3 Semantic word2vec Similarity Features

Every text snippet T has a set of words. Each word has a fixed-length word embedding representation, $w \in \mathbb{R}^d$, where d is the dimensionality of the word embedding. Thus for a text snippet T we define $T = \{w_1, \cdots, w_k\}$, where k is the number of words in T. The word embedding representation is computed offline following Mikolov et al. approach (Mikolov et al., 2013).

To compute similarity scores, we represent each text snippet by a feature vector; different alternatives for feature representations are adopted as described next.

2.3.1 Average Word Embedding Similarity

For a text snippet T that has k words, we compute the average vector as follows:

$$T^{\mu} = \frac{\sum_{i=1}^{k}(w_i)}{k} \quad (6)$$

Notice that $T^{\mu} =\in \mathbb{R}^d$. This leads to the following cosine similarity feature.

$$CS_{\mu}(T_1, T_2) = \frac{\vec{T_1^{\mu}} \cdot \vec{T_2^{\mu}}}{||\vec{T_1^{\mu}}|| \, ||\vec{T_2^{\mu}}||} \quad (7)$$

2.3.2 Covariance Word Embedding Similarity

Instead of computing the average vector, we can compute a covariance matrix $C \in \mathbb{R}^{d \times d}$. The covariance matrix C is computed by treating each dimension as a random variable and every entry in $C_{u,v}$ is the covariance between the pair of variables (u, v). The covariance between two random variables u and v is computed as in eq. 8, where k is the number of observations (words).

$$C_{u,v} = \frac{\sum_{i=1}^{k}(u_i - \bar{u})(v_i - \bar{v})}{k - 1} \quad (8)$$

The matrix $C \in \mathbb{R}^{d \times d}$ is a symmetric matrix. We compute a vectorized representation of the matrix C as the stacking of the lower triangular part of matrix C as in eq. 9. This process produces a vector $T^{Cov} \in \mathbb{R}^{d \times (d+1)/2}$

$$T^{Cov} = \textbf{vect}(C) = \{C_{u,v} : u \in \{1, \cdots, d\}, v \in \{u, \cdots, d\}\} \quad (9)$$

This leads to the following cosine similarity feature.

$$CS_{Cov}(T_1, T_2) = \frac{\vec{T_1^{Cov}} \cdot \vec{T_2^{Cov}}}{||\vec{T_1^{Cov}}|| \, ||\vec{T_2^{Cov}}||} \quad (10)$$

2.4 Ranking Using SVM

Although Subtask D is a re-ranking task, it has also a classification task where answers need to be ranked and labeled with either *true* or *false*; the former designates a *Direct* or *Relevant* answer to the new question, and the latter designates an *Irrelevant* answer. In our last year's submission (Malhas et al., 2016) we used learning-to-rank module for re-ranking pairs, but we used a simple heuristic to give labels to the candidate question-answer pairs. This year we use SVM to give a label for every candidate pair using the SVM model. In addition to labeling pairs, we use the decision scores from the SVM model for re-ranking the candidate question-answer pairs.

3 Experimental Evaluation

In this section we present the experimental setup and results of our primary, contrastive-1 and contrastive-2 submissions.

3.1 Experimental Setup

We used the Arabic collection of questions and their potentially related question-answer pairs provided by Task 3 organizers to train our word embedding model. The Gensim[3] tool was used to generate the word2vec model from training data[4], setting $d = 100$. We used the learned model to compute our features as described in section 2. Features were generated for the three data setups described in section 2.1.

3.2 Submissions and Results

The differences among our submitted runs is based on the selection of the features. In all cases we use linear SVM for classifying and ranking question-answer pairs. Details on our official submissions

[3] http://radimrehurek.com/gensim/

[4] Testing data are held out during the computation of the word2vec model.

	MAP	AvgRec	MRR	P	R	F1	Acc
QU-BigIR Contrastive 2	59.48	83.83	64.56	55.35	70.95	62.19	66.15
QU-BigIR Contrastive 1	59.13	83.56	64.68	49.37	85.41	62.57	59.91
QU-BigIR Primary	56.69	81.89	61.83	41.59	70.16	52.22	49.64
Baseline 1 (IR)	60.55	85.06	66.08	-	-	-	-
Baseline 2 (Random)	48.48	73.89	53.27	39.04	66.43	49.18	46.13
Baseline 3 (all true)	-	-	-	39.23	100	56.36	39.23
Baseline 4 (all false)	-	-	-	-	-	-	60.77

Table 1: The official scores attained by our primary and contrastive submissions to SemEval 2017 Task 3-SubTask D

are summarized next. Table 1 presents the official results of our submissions.

Contrastive-1. We use the set of term-based similarity features defined in section 2.2. We compute these features for all data setups defined in section 2.1. This results in 18 features in total. Six features for every data setup **(QQ, QA and QQA)**. We tuned the parameter C for the linear SVM on the development set.

Contrastive-2. In addition to the features used in Contrastive-1 submission, we use the set of semantic word2vec based similarity features defined in section 2.3. This results in 24 features in total; eight features for every data setup **(QQ, QA and QQA)**. This submission produced our best MAP results.

Primary. We use the full set of similarity features defined in section 2.2 and section 2.3. In addition, we performed a weighted score fusion with an SVM model based on fixed length representation using Covariance word embedding. The feature vectors we used are computed using equation 9. We tuned the model weights using the development set. Table 2 shows that this setup gets the best MAP results on the development set.

	MAP
QU-BigIR Contrastive 1	42.54
QU-BigIR Contrastive 2	42.87
QU-BigIR Primary	43.41
Baseline (Random)	29.79

Table 2: The development set MAP scores obtained by our primary and contrastive submissions.

3.3 Discussion

- Our best official submission is Contrastive-2 using both term-based similarity features and semantic word2vec similarity features. This indicates that the two similarity features types are complementing each other.

- Our results justify the usage of SVM model for labeling and re-ranking question-answer pairs. This is clear in the **P, R, F1** and **Acc** scores reported across all other baselines. We report very competitive MAP scores to the best performing ranking systems which are not using any form of labeling such as IR-baseline.

- Score fusion in our primary run did not achieve best results on the official test set while it was the best run in our experiments on the development set. We believe that this happened due to the difference in the source of question-answer pairs in the development set compared to the the official test set where the test set contains only medical questions.

4 Conclusion

This paper describes the system we developed to participate in SemEval-2017 Task 3 on Community Question Answering. Our system has focused on the Arabic Subtask D which is confined to Answer Selection in Community Question Answering, i.e., finding good answers for a given new question.

We have adopted a supervised learning approach where linear SVM models were trained over similarity features. In our best submission, term-based similarity features and word2vec similarity features were both used; our system ranked second among the other participating teams.

Acknowledgments

This work was made possible by NPRP grant# NPRP 6-1377-1-257 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

Rana Malhas, Marwan Torki, and Tamer Elsayed. 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-2016)*. Association for Computational Linguistics, San Diego, California, pages 866–871.

Donald Metzler and Tapas Kanungo. 2008. Machine learned sentence selection strategies for query-biased summarization. In *SIGIR Learning to Rank Workshop*. pages 40–47.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

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*. Association for Computational Linguistics, Vancouver, Canada, SemEval '17.

Anna Shtok, Gideon Dror, Yoelle Maarek, and Idan Szpektor. 2012. Learning from the past: answering new questions with past answers. In *Proceedings of the 21st international conference on World Wide Web*. ACM, pages 759–768.

Reem Suwaileh, Mucahid Kutlu, Nihal Fathima, Tamer Elsayed, and Matthew Lease. 2016. Arabicweb16: A new crawl for today's arabic web. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '16, pages 673–676.

Liu Yang, Qingyao Ai, Damiano Spina, Ruey-Cheng Chen, Liang Pang, W. Bruce Croft, Jiafeng Guo, and Falk Scholer. 2016. Beyond factoid qa: Effective methods for non-factoid answer sentence retrieval. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings*, Springer International Publishing, pages 115–128.