QU-IR at SemEval 2016 Task 3: Learning to Rank on Arabic Community Question Answering Forums with Word Embedding

Rana Malhas and **Marwan Torki** and **Tamer Elsayed**
Qatar University
Department of Computer Science & Engineering
Doha, Qatar
{rana.malhas,mtorki,telsayed}@qu.edu.qa

Abstract

Resorting to community question answering **(CQA)** websites for finding answers has gained momentum in the past decade with the explosive rate at which social media has been proliferating. With many questions left unanswered on those websites, automatic and smart question answering systems have seen light. One of the main objectives of such systems is to harness the plethora of existing answered questions; hence transforming the problem to finding good answers to newly posed questions from similar previously-answered ones. As SemEval 2016 Task 3 "Community Question Answering" has focused on this problem, we have participated in the Arabic Subtask. Our system has adopted a supervised learning approach in which a learning-to-rank model is trained over data (questions and answers) extracted from Arabic CQA forums using word2vec features generated from that data. Our primary submission achieved a 29.7% improvement over the MAP score of the baseline. Post submission experiments were further conducted to integrate variations of the word2vec features to our system. Integrating covariance word embedding features has raised the the improvement over the baseline to 37.9%.

1 Introduction

The ubiquitous presence of community question answering (CQA) websites has motivated research in the direction of 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. The ranking functionality is vital to push away irrelevant and low quality answers, which is commonplace in CQA as they are generally open with no restrictions on who can post or answer questions.

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 واالتشخيص 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 ولبس 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 QApairs

To this effect, SemEval 2016 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

Proceedings of SemEval-2016, pages 791–796,
San Diego, California, June 16-17, 2016.

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.

The Arabic training, development and test datasets provided by the organizers were extracted from Arabic medical forums (webteb[1], altibbi[2]), and "Consult Islamweb"[3]. Further details about SemEval 2016 Task 3 can be found in (Nakov et al., 2016).

In this paper, we describe the system we developed to participate in SemEval-2016 Task 3 (Arabic Subtask). The system has leveraged a supervised learning approach over word2vec features extracted from a collection of questions and their candidate question-answer pairs to build a ranking model. The functionality of the developed system is confined to the answers re-ranking task described by SemEval 2016 Task 3. With the MAP (Mean Average Precision) being the official measure for evaluation, our efforts were mainly focused on optimizing this measure. Our developed system has achieved a MAP score improvement of 29.7% over the baseline via our primary submission, and an improvement of 37.9% via our post-submission enhancements and experiments.

The rest of the paper is organized as follows; the approach and the generated features are introduced in section 2; the experimental evaluation and setup followed by our submissions to the Arabic Subtask and their results are presented in section 3. Enhancements and further experiments conducted are also presented in section 3 before concluding with final remarks.

2 Approach

We tackled the answer ranking task with a supervised learning approach that leveraged learning-to-rank models. The features used in training are mainly semantic, where vectorized word embedding representations were used as features in different alternatives, as explained below. An overview of our system is depicted in Figure 2. Details regarding the specific models and features used in our primary and contrastive submissions are presented in section 3.

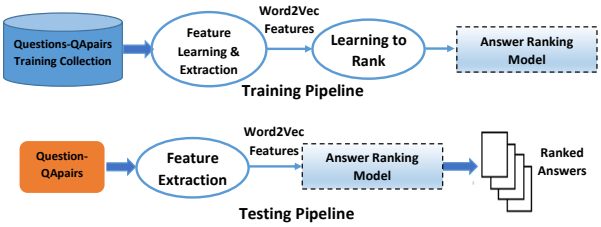

Figure 2: System overview

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 document according to three setups:

- **QQA:** We consider the concatenation of an original question q and one **pair** p of its associated question-answer pairs as a document d.
- **QA:** We consider the concatenation of an original question q and one **answer** of its associated question-answer pairs as a document d.
- **QQ:** We consider the concatenation of an original question q and one **question** of its associated question-answer pairs as a document d.

We have extracted features for the above data setups seeking those with the most discriminating power against our ranking problem; this is further elaborated in section 3.

2.2 Features

Every document $d_n \in D$, where $n \in \{1, \cdots, N\}$, has a set of words. Each word has a fixed-length word embedding representation, $w \in \mathbb{R}^{Dim}$, where Dim is the dimensionality of the word embedding. Thus for every document d_n in the set D, we define $d_n = \{w_1, \cdots, w_{k_n}\}$, where k_n is the number of words in the document d_n. The word embedding representation is computed offline following Mikolov et al approach (Mikolov et al., 2013).

To enable learning, we represent each document by a feature vector; different alternatives for feature representations are adopted as described next.

[1] https://www.webteb.com/
[2] http://www.altibbi.com/
[3] http://consult.islamweb.net/mainpage/index.php

2.2.1 Average Word Embedding

For a document that has k_n words, we compute the average vector as follows:

$$\mu_n = \frac{\sum_{i=1}^{k_n}(w_i)}{k_n} \quad (1)$$

Notice that $\mu_n \in \mathbb{R}^{Dim}$. For our primary and contrastive submissions, we only used the average vector μ_n to represent its respective document.

2.2.2 Covariance Word Embedding

Instead of computing the average vector, we can compute a covariance matrix $C \in \mathbb{R}^{Dim \times Dim}$. The covariance matrix C is computed by treating each dimension as a random variable and every entry in $C_{n_{u,v}}$ is the covariance between the pair of variables (u, v). The covariance between two random variables u and v is computed as in eq. 2, where k_n is the number of observations (words).

$$C_{n_{u,v}} = \frac{\sum_{i=1}^{k_n}(u_i - \bar{u})(v_i - \bar{v})}{k_n - 1} \quad (2)$$

The matrix $C_n \in \mathbb{R}^{Dim \times Dim}$ is a symmetric matrix. We compute a vectorized representation of the matrix C_n as the stacking of the lower triangular part of matrix C_n as in eq. 3. This process produces a vector $v_n \in \mathbb{R}^{Dim \times (Dim+1)/2}$

$$\textbf{vect}(C_n) = \left\{ C_{n_{u,v}} : u \in \{1, \cdots, Dim\}, v \in \{u, \cdots, Dim\} \right\} \quad (3)$$

In our post submission experiments we used the covariance descriptors v_n in comparison to the basic μ_n average vectors.

2.2.3 Unigrams and tf-idf weighting

In our post submission experiments, we also used the standard unigram representation with tf-idf weighting. We have chosen the most frequent 5000 unigrams from the training data to represent every document as a sparse vector of 5000 dimensions.

2.3 Ranking Models

A learning-to-rank (L2Rank) setup was adopted that is similar to (Chen et al., 2015; Surdeanu et al., 2008). The L2Rank models were trained over the labeled Arabic data provided by the SemEval 2016 Task 3 organizers. The data constituted 1,031 original questions and their potentially related 30,411 question-answer pairs (QApairs); i.e. about 30 QApairs per original question. Each QApair is labeled by either being *Direct*, *Relevant* or *Irrelevant* with respect to the original question; the distribution of these labels are 3.0%, 57.0% and 40.0%, respectively (Nakov et al., 2015).

Two algorithms were used to train our learning-to-rank models, namely the MART (Multiple Additive Regression Trees, a.k.a. Gradient boosted regression tree) algorithm and the Random Forests algorithm.

3 Experimental Evaluation

In this section we present the experimental setup and results of our primary, contrastive-1 and contrastive-2 submissions, in addition to our post-submission experiments.

3.1 Experimental Setup

We have used the Arabic collection of questions and their potentially related question-answer pairs provided by Task3 organizers for training our models. We evaluated those models using the development dataset of 7,355 question-QApairs instead of the full provided dataset of 7,385 question-QApairs; one question (out of the 250) and its potentially related 30 QApairs were not properly formed and thus excluded. Another data preprocessing step was to parse and transform the XML files into flat files for easier data processing/tracking of question-answers.

The Gensim[4] tool was used to generate the word2vec model from training data[5]. We used the learned model to compute our features as described in section 2.2. Features were generated for the three data setups described in section 2.1.

RankLib[6] was used to create and evaluate our learning-to-rank models. Although we have experimented with a number of pairwise and listwise learning-to-rank algorithms, we adopted pointwise L2Rank algorithms in our submissions as they exhibited a relatively better performance than the other two categories.

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

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

[6] http://sourceforge.net/p/lemur/wiki/RankLib/

3.2 Evaluation Measures

Since MAP (Mean Average Precision) is the official evaluation measure to evaluate Task 3-Subtask D submissions, we focused our experiments and evaluations to optimize this measure. Time constraints have withheld us from optimizing the other evaluation measures that are also adopted by the task's official scorer, such as F1 measure, accuracy, etc.

3.3 Submissions and Results

Table 1 below summarizes the official results of our submissions which are discussed in the context of the following sub-sections.

3.3.1 Primary and Contrastive Submissions

The average word embedding features were used in all three submissions. We used $Dim = 100$ for the word2vec model we learned from training data. The differences among submissions lie in the data setup (QQA, QA or QQ) used in generating the features; and the algorithm deployed in training the L2Rank model that ranks the answers.

Submission	MAP
Primary	38.63
Contrastive-1	37.80
Contrastive-2	39.07
Baseline	29.79

Table 1: The official MAP scores attained by our primary and contrastive submissions to SemEval 2016 Task 3-SubTask D

Primary submission. The QQA data setup was used in generating the average word embedding features; and the MART algorithm (Multiple Additive Regression Trees, a.k.a. Gradient boosted regression tree) was used to train the L2Rank model. This submission has attained a MAP score of 38.63 which placed it in the fourth position among the other primary submissions of other participating teams. It achieved a 29.7% improvement over the baseline (29.79).

Contrastive-1 submission. The QA data setup was used in generating the features; and the Random Forests algorithm was deployed in training the L2Rank model. This submission has attained a MAP score of 37.80 which is lower than our other two submissions (Table 1). This suggests that using the QA data setup in feature generation (i.e. using the original question and **answers** of the QApairs while leaving out the questions of the QApairs), is not as good as using the QQA data setup (i.e. using the original questions and their **QApairs**). Further experiments might be needed to assert this finding.

Contrastive-2 submission. The QQA data setup was used in generating the features, and again the Random Forests algorithm was deployed in training the L2Rank model. Another difference in the setup of this submission was using 20% of the training data in validating the trained model while using the remaining 80% for training. This submission has performed better than our primary and contrastive-1 submissions; it attained a MAP score of 39.07 and an improvement of 31.3% over the baseline.

It is worth noting that feature values in the primary and contrastive-1 submissions were normalized using zscore and sum, respectively. In the primary submission, each feature was normalized by its *mean/standard deviation*, while in the contrastive-1 submission, each feature was normalized by the *sum* of all its values.

Although Subtask D at the surface is a re-ranking task, it has also embedded a classification task where answers need to be ranked and labeled with either *true* or *false*; the former designates a *Direct* or *Relevant* answer, and the latter designates an *Irrelevant* answer. In all submissions, we have adopted a simple heuristic of labeling the top 10 ranked answers with the label *true*, and the remaining answers with *false* otherwise. Alternatively, a supervised classifier can be used to predict the answer labels, or find a good cutoff threshold point (such as the average or median of answers rank scores) to label those exceeding that threshold with *true*, and *false* otherwise.

3.3.2 Post-Submission Experiments

Further experiments were conducted to explore the performance of Covariance Word Embedding (CovWE) and unigram features as compared to the Average Word Embedding (AvgWE) features. In Table 2, we report the MAP scores achieved by these features using a dimensionality of 50 and 100, respectively, for representing the vectors of word embeddings. In our post-submission experiments, we only extracted features using the QQA data setup.

As such, we only include the results of our primary and contrastive-2 submissions in Table 2 because their features were also extracted using the QQA data setup, unlike the contrastive-1 submission.

Experiment/Features	Normalization	MAP
AvgWE-50	-	36.01
AvgWE-100 (primary)	Zscore	38.63
AvgWE-100 (contrastive-2)	-	39.07
AvgWE-100 and Unigrams	-	37.71
CovWE-50	Linear	**41.07**
CovWE-100	Linear	**40.68**
CovWE-100 and Unigrams	-	37.51
Baseline	-	29.79

Table 2: Post-submission experiments comparing the performance of Covariance Word Embedding (CovWE) features and Unigrams to that of Average Word Embedding (AvgWE) features. The suffix numbers 50 and 100 designate the dimensionality of the vectors representing the word embeddings. Best scoring features are boldfaced.

In most of the experiments reported in Table 2, the MART algorithm was used for training the L2Rank models; whereas, for the AvgWE-100 and Unigrams experiment, the Random Forests algorithm was used. In general, the MART algorithm performed better in the majority of our experiments that we have conducted but have not reported. The main observations worth mentioning regarding the experiments in Table 2 are:

- Using tf.idf weighted uni-grams of the most frequent 5000 words along with word2vec features (AvgWE and CovWE) did not mark an improvement over using word2vec features solely.
- Normalization of feature values seem to have a tendency of enhancing the achieved MAP scores when applied. For this reason, we include in Table 2 the normalization scheme (if any) that was adopted in each experiment.
- The discriminant potential of the covariance word embedding features seem to be relatively stronger than that of average word embedding features. For example, the features CovWE-50 and CovWE-100 have achieved relatively higher MAP scores than AvgWE-50 and AvgWE-100, respectively. With their 41.07 and 40.68 MAP scores, CovWE features have achieved an improvement of about 37.9% over the baseline. AvgWE-100 and AvgWE-50 followed with the MAP scores of 38.63 (primary submission score) and 36.01, respectively; hence, attaining lower improvements (29.7% and 20.9%) over the baseline.
- The covariance word embedding features CovWE-50 and CovWE-100 have attained comparable MAP scores of 41.07 and 40.68, respectively. Interestingly, the CovWE-50 experiment consumed 44.5 minutes to learn the L2Rank model, while the CovWE-100 experiment consumed 5.27 hours. This finding is also suggesting that covariance word embedding features seem to have a relatively higher discriminating potential even with lower dimensions.

More rigorous benchmarking experiments might be needed to further verify the merit of the above implications.

4 Conclusion

This paper describes the system we have developed to participate in SemEval-2016 Task 3 on Community Question Answering. Our system has focused on the Arabic Subtask which is confined to Answer Selection in Community Question Answering, i.e. finding good answers for a given new question. The training data provided by the organizers were extracted from Arabic medical forums (*webteb* and *altibbi*) and *consult islamweb*.

We have adopted a supervised learning approach where learning-to-rank models were trained over word2vec features generated from the training data. In our primary submission, average word embedding features were used; our system ranked fourth among the other participating teams. It achieved a 29.7% improvement over the baseline. Post-submission experiments were further conducted to enhance the system and integrate covariance word embedding features. The enhanced system marked an improvement of 37.9% over the baseline.

Our experiments have provided preliminary evidence regarding the discriminant potential of the

covariance word embedding features over the average word embedding features; the former type of word2vec features enabled the learned model to attain a relatively better MAP score.

Furthermore, the highly comparable MAP scores attained by the covariance word embedding features for 50 and 100 dimensions (Table 2) suggest another interesting finding: the covariance word embedding features seem to have a relatively higher discriminating potential even with lower dimensions.

In future work, we intend to integrate more semantic features extracted from richer and larger semantic Arabic resources.

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).

References

Ruey-Cheng Chen, Damiano Spina, W Bruce Croft, Mark Sanderson, and Falk Scholer. 2015. Harnessing semantics for answer sentence retrieval. In *Proceedings of the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 21–27. ACM.

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, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. 2015. SemEval-2015 task 3: Answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '2015, pages 269–281, Denver, Colorado, June. Association for Computational Linguistics.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. 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.

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*, pages 759–768. ACM.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online qa collections. In *ACL*, volume 8, pages 719–727.