ECNU at SemEval-2017 Task 3: Using Traditional and Deep Learning Methods to Address Community Question Answering Task

Guoshun Wu[1], **Yixuan Sheng**[1], **Man Lan**[1,2*], **Yuanbin Wu**[1,2]
[1]Department of Computer Science and Technology,
East China Normal University, Shanghai, P.R.China
[2]Shanghai Key Laboratory of Multidimensional Information Processing
51141201064,51164500026@stu.ecnu.edu.cn
mlan,ybwu@cs.ecnu.edu.cn

Abstract

This paper describes the systems we submitted to the task 3 (Community Question Answering) in SemEval 2017 which contains three subtasks on english corpora, i.e., subtask A: Question-Comment Similarity, subtask B: Question-Question Similarity, and subtask C: Question-External Comment Similarity. For subtask A, we combined two different methods to represent question-comment pair, i.e., supervised model using traditional features and Convolutional Neural Network. For subtask B, we utilized the information of snippets returned from Search Engine with question subject as query. For subtask C, we ranked the comments by multiplying the probability of the pair "related question – comment" being Good by the reciprocal rank of the related question.

1 Introduction

The purpose of Community Question Answering task in SemEval 2017 (Nakov et al., 2017) is to provide a platform for finding good answers to new questions in a community-created discussion forum, where the main task (subtask C) is defined as follows: given a new question and a large collection of question-comment threads created by a user community, participants are required to rank the comments that are most useful for answering the new question. Obviously, this main task consists of two optional subtasks, i.e., Question-Comment Similarity (subtask A, also known as *answer ranking*), which is to re-rank comments/answers according to their relevance with respect to the question, and Question-Question Similarity (i.e., subtask B, also known as *question retrieval*), which is to retrieve the similar questions according to their semantic similarity with respect to the original question. More, a new subtask: Multi-Domain Duplicate Detection Subtask (i.e., subtask E) which is to identify duplicate questions in StackExchange has been added to SemEval 2017 task 3.

To address subtask A, we explored a traditional machine learning method which uses multiple types of features, e.g., Word Match Features, Topic Model-based Features, and Lexical Semantic Similarity Features. Additionally, for subtask A, we also built a Convolutional Neural Network (CNN) model to learn joint representation for question-comment (Q-C) pair. For subtask B, we utilized the information of snippets returned from Search Engine with question subject as query, e.g., we counted the frequency of each word in each snippets list and added the words which appear in the subject of original question and the frequency is more than 1 to the subject of related question. Since subtask C can be regarded as a joint work of the two above-mentioned subtasks, we ranked the comments by multiplying the probability of the pair "related question – comment" being Good by the reciprocal rank of the related question. As for subtask E, we did not submit the results because of the large amount of dataset.

The rest of this paper is organized as follows. Section 2 describes our system. Section 3 describes experimental setting. Section 4 and 5 report results on training and test sets. Finally, Section 6 concludes this work.

2 Systems Description

For subtask A, we presented two different methods i.e., using traditional linguistic features and learning a CNN model to represent question and comment sentences. For subtask B, besides Word

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 365–369,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

Match, Topic Model based, and Lexical Semantic Similarity features, we also extracted Search Engine Extensional feature. For subtask C, we ranked the comments by multiplying the probability of the pair “relevant question – comment” being Good by the reciprocal rank of the related question.

2.1 Features Engineering

All three subtasks can be regarded as an estimation task of sentence semantic measures which can be modeled by various types of features. Besides Word Match, Topic Model Based, Lexical Semantic Similarity, and Comment Information Features used in our previous work (Wu and Lan, 2016), we also extract three types of novel features, i.e., Meta Data Features, Google Ranking Feature, and Search Engine Extensional Features. The details of features are described as follows. Here we took the *Q-Q* pair for example.

Word Matching Feature (WM): Inspired by the work of (Zhao et al., 2015), we adopt word matching feature in our system. This feature represents the the proportions of co-occurred words that between a given sentence pair. Given a *Q-Q* pair, this feature is expressed in the following nine measures:$|Q_0 \cap Q_1|, |Q_0 \cap Q_1|/|Q_0|, |Q_0 \cap Q_1|/|Q_1|, |Q_1 - Q_0|/|Q_1|, |Q_0 - Q_1|/|Q_0|, |Q_0 \cap Q_1|/|Q_0 - Q_1|, |Q_0 \cap Q_1|/|Q_1 - Q_0|, |Q_0 \cap Q_1|/|Q_0 \cup Q_1|, 2*|Q_0 \cap Q_1|/(|Q_0|+|Q_1|)$, where $|Q_0|$ and $|Q_1|$ are the number of the words of Q_0 and Q_1.

Topic Model based Feature (TMB): Topic model based feature has been proved beneficial for question retrieval and answer ranking tasks by the work of (Duan et al., 2008; Qin et al., 2009). We use the *GibbsLDA++* (Phan and Nguyen, 2007) Toolkit with 100,000 random sampling question and answer pairs from Qatar Living data to train the topic model. In training and test phase, Q_0 and Q_1 are transformed into an 100-dimensional topic-based vectors using pre-trained topic model. After that we calculate the cosine similarity, Manhattan distance and Euclidean distance between these two vectors and regard the scores as TMB feature. Inspired by the work of (Filice et al., 2016), we also adopt four kinds of nonlinear kernel functions to calculate the distance between two vectors, i.e., ”polynomial”, ”rbf”, ”laplacian” and ”sigmoid”.

Lexical Semantic Similarity Feature (LSS): Inspired by (Yih et al., 2013a), we included the lexical semantic similarity feature in our model. Two types of 300-dimensional vectors are pre-trained on Qatar Living data with word2vec (Yih et al., 2013b) and Glove (Pennington et al., 2014) toolkits. We select the maximum, minimum and average values for each dimension of words vectors to make up a vector to represent the sentence. After obtained the vector representation of Q_0 and Q_1, we also calculated the nine distance measures mentioned in **TMB**.

Note that all above three types of features are adopted in both answer ranking and question retrieval tasks.

Search Engine Extensional Feature (SEE):

We first got two lists of 10 snippets returned by search engine (i.e., Google, Bing) with the subjects of original question Q_0 and related question Q_1 as query. Then we counted the frequency of each word in each snippets list and added the words which appear in the Q_1/Q_0 and the frequency is more than 1 to the subject of Q_0/Q_1. Finally, the *WM* features are calculated based the changed subjects of Q_0 and Q_1.

Google Ranking Feature (GR): The reciprocal rank of the related question as given by Google is regarded as one dimensional feature.

Meta Data Feature (MD): Meta data is often helpful for finding good answers and question category distribution of user posted answers is an important meta data information. There are 28 question categories in the training data, we calculate the following values as features, i.e., the numbers of answers answered by all users in a certain category and the numbers of answers answered by a single user in all categories are normalized using max-min scaling, forming two 28-dimensional vectors. We also take the quality (i.e., Good, PotentiallyUseful, and Bad) of answers into consideration. The numbers of different quality answers answered by all users under a category and the numbers of different quality answers answered by a users in all categories are normalized using max-min scaling, forming two 3*28-dimensional vectors.

Comment Information Feature (CI):

We also extracted following comment information features to measure the informativeness of a comment text: (1) comment unigram feature, we constructed a vocabulary with the words appeared more than twice in the training data, generating a 9000-dimensional vector of one-hot form for each comment. (2) comment ner feature, we extracted nine types of name entity information in the comment, i.e., ”Duration”, ”Location”, ”Person”, ”Organization”, ”Percent”, ”Ordinal”, ”Time”, ”Date”, and ”Money” with the *CoreNLP* tool, generating a nine-dimensional one-hot forming vector. (3) comment special characters feature, We extracted the following five special characters features from the comment, i.e., email, url, ”@”, ”...”, and ”?”, generating a 5-dimensional vector of one-hot form for every comment.

Note that **MD** and **CI** features are used in answer ranking task only. **GR** and **SEE** features are used in question retrieval task only.

2.2 CNN to address subtask A

We proposed a convolutional neural network to model question-comment sentence. As illustrated in Figure 1, it first takes the embeddings (here we used 300-dimensional Glove vectors) (Pennington et al., 2014) of question and comment words as inputs and then summarizes the meaning of question and comment through convolution and pooling. Finally the softmax output of *Good* classes is regarded as ranking score between question and comment by a simple hidden layer building on the concatenation of two feature vectors and softmax operation. For CNN model, we set the filter numbers as 1,2,3 and 4 with same feature map of 100 and the stochastic gradient descent algorithm is used to update the parameters with learning rate of 0.001 and cross entropy as loss function.

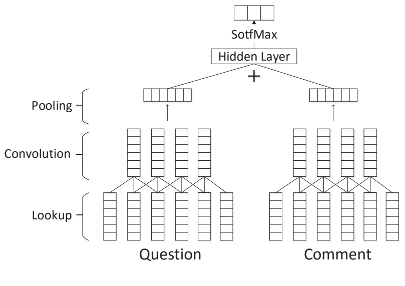

Figure 1: An illustration of CNN for question-comment similarity estimation.

3 Experimental Setting

3.1 Datasets

Table 1 shows the statistics of training, development, test data sets of SemEval 2016 and test data sets of SemEval 2017, where the #_ original, #_ related, and #_ answers represent the number of original questions, related questions and answers, respectively. The types of comments with respect to original question and related question fall into three classes: $Good$, $PotentiallyUseful$ and Bad. The types of related question with respect to original question fall into three classes: $PerfectMatch$, $Relevant$ and $Irrelevant$.

Subtask	Data	#_original	#_related	#_answers
A	train	–	5,898	37,848
	dev	–	500	5,000
	2016 test	–	327	3,270
	2017 test	–	293	2,930
B	train	267	2,669	26,690
	dev	50	500	5,000
	2016 test	70	700	7,000
	2017 test	88	880	8,800
C	train	267	2,669	26,690
	dev	50	500	5,000
	2016 test	70	700	7,000
	2017 test	88	880	8,800

Table 1: Statistics of datasets.

3.2 Preprocessing

Firstly, we removed stop words and punctuation, and changed words to their lowercase. After that, we performed tokenization and stemming using NLTK[1] Toolkit.

3.3 Learning Algorithm

We compared various machine learning algorithms such as Logistic Regression, Random Forest and AdaBoost implemented by SKLearn[2] with default parameters setting for their good performance in preliminary experiments. The probabilistic scores of $PerfectMatch$ and $Good$ classes returned by classifiers are regarded as ranking scores of question-question pair and question-comment pair. According to their performances with diverse features in three subtasks, they are used in different subtasks in our final submitted results.

4 Experiments on Training Data

4.1 Results on Subtask A

Table 2 shows the results of subtask A with two different methods on SemEval 2016 Test data sets.

[1] http://www.nltk.org/

[2] http://scikit-learn.org/stable/

Methods	Features	Test MAP(%)
Traditional NLP Features	All	77.82
	All - WM	76.60
	All - TMB	77.46
	All - MD	73.53
	All - CI	76.56
	All - LSS	76.43
CNN	–	77.76
Tra + CNN	–	**79.30**

Table 2: Results of subtask A with two different methods. ”All” means to all features and ”-” means to exclude some feature groups.

4.2 Results on Subtask B

Table 3 summarizes the results of subtask B on SemEval 2016 Test data sets with different features and algorithms.

Features	Algorithms		
	LR	AdaBoost	RandomForest
All	**75.43**	75.14	74.85
All - WM	74.31	74.78	74.14
All - GR	71.33	73.43	71.33
All - TMB	74.34	74.65	74.25
All - SEE	72.34	73.65	74.10
All - LSS	73.65	74.51	74.21

Table 3: Results of subtask B.

4.3 Results on Subtask C

Table 2 shows the results of subtask C with different algorithms and features on SemEval 2016 Test data sets.

Features	Algorithms		
	AdaBoost	Random Forest	LR
All	52.04	50.89	48.39
All - WM	51.70	50.63	47.59
All - TMB	51.82	49.05	47.59
All - MD	**52.35**	50.90	49.12
All - CI	49.19	48.93	46.75
All - LSS	50.54	49.48	47.73

Table 4: Results of subtask C.

4.4 Conclusion on Experimental results

Based on above experimental results, we find that

(1) For subtask A, all the features (e.g., WM, TMB, MD, CI and LSS) make contribution to the improvement of performance. The CNN based model achieves comparable performance with traditional method and with the average value of scores returned by two methods as ranking score achieves the best performance.

(2) For subtask B, three algorithms such as Logistic Regression, AdaBoost and Random Forest achieve comparable results with traditional NLP features. Specially, LR with all features achieve the best performance.

(3) For subtask C, AdaBoost with all features (excluding MD feature) makes the best result compared with Random Forest and Logistic Regression.

4.5 Systems Configuration

Based on above experimental analysis, the three system configurations on SemEval 2017 Test data sets are listed as followings:

(1) subtask A: We used the combination of traditional method and CNN as primary run. Traditional method and CNN serve as contrastive1 run and contrastive2 run.

(2) subtask B: Logistic Regression with all NLP features is used as primary run. AdaBoost and Random Forest with all NLP features are used as contrastive1 run and contrastive2 run.

(3) subtask C: AdaBoost with all NLP features is used as primary run in the test set. Random Forest and Logistic Regression with all NLP features are used as contrastive1 run and contrastive2 run.

5 Results on 2017 Test Data

Table 5 shows the results on SemEval 2017 test set which are released by the organizers.

subtask	run(rank)	MAP(%)
A	ECNU-primary(4)	86.72
	ECNU-contrastive1	86.78
	ECNU-contrastive2	83.15
	Kelp-primary(1)	88.43
B	ECNU-primary(11)	41.37
	ECNU-contrastive1	42.37
	ECNU-contrastive2	42.48
	simbow-primary(1)	47.22
C	ECNU-primary(5)	10.54
	ECNU-contrastive1	10.54
	ECNU-contrastive2	13.29
	IIT-UHH-primary(1)	15.46

Table 5: Our results and the best results on three subtasks test sets. The numbers in the brackets are the official ranking.

From the results, we find: (1) In subtask A, the combination of two methods does not make obvious contribution and the CNN based method has a certain gap with traditional method, which is inconsistent with the results on training data as our expectation. (2) In subtask B, the result using LR does not make expected result compared with AdaBoost and Random Forest algorithms. (3) In subtask C, beyond our expectation, the method using LR algorithm achieved the best result.

6 Conclusion

In this paper, we proposed multiple strategies (i.e., traditional method of extracting features and deep learning models) to address Community Question Answering task in SemEval 2017. For subtask A, we train a classifier and learn the question-comment representation based CNN. For subtask B, we we utilized the information of snippets searching from Search Engine with question as query. For subtask C, We ranked the comments by multiplying the probability of the pair “relevant question – comment” being Good by the reciprocal rank of the related question.

Acknowledgments

This research is supported by grants from Science and Technology Commission of Shanghai Municipality (14DZ2260800 and 15ZR1410700), Shanghai Collaborative Innovation Center of Trustworthy Software for Internet of Things (ZF1213) and NSFC (61402175)

References

Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. In *ACL*, pages 156–164.

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. 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-2016)*, pages 1116–1123, San Diego, California, June. Association for Computational Linguistics.

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.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543.

Xuan-Hieu Phan and Cam-Tu Nguyen. 2007. Gibbslda++: Ac/c++ implementation of latent dirichlet allocation (lda).

Zengchang Qin, Marcus Thint, and Zhiheng Huang. 2009. Ranking answers by hierarchical topic models. In *Next-Generation Applied Intelligence*, pages 103–112. Springer.

Guoshun Wu and Man Lan. 2016. Ecnu at semeval-2016 task 3: Exploring traditional method and deep learning method for question retrieval and answer ranking in community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 872–878, San Diego, California, June. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013a. Question answering using enhanced lexical semantic models.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013b. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1744–1753, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jiang Zhao, Man Lan, and Jun Feng Tian. 2015. Ecnu: Using traditional similarity measurements and word embedding for semantic textual similarity estimation. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 117–122, Denver, Colorado, June. Association for Computational Linguistics.