HITSZ-ICRC: Exploiting Classification Approach for Answer Selection in Community Question Answering

Yongshuai Hou, Cong Tan, Xiaolong Wang
Yaoyun Zhang, Jun Xu and Qingcai Chen
Key Laboratory of Network Oriented Intelligent Computation
Department of Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School
HIT Campus, The University Town of Shenzhen, Shenzhen, 518055, China
{yongshuai.hou, viptancong}@gmail.com, wangxl@insun.hit.edu.cn
{xiaoni5122, hit.xujun,qingcai.chen}@gmail.com

Abstract

This paper describes the participation of the HITSZ-ICRC team on the Answer Selection Challenge in SemEval-2015. Our team participated in English subtask A, English subtask B and Arabic task. Two approaches, ensemble learning and hierarchical classification were proposed for answer selection in each task. Bag-of-words features, lexical features and non-textual features were employed. For the Arabic task, features were extracted from both Arabic data and English data that translated from the Arabic data. Evaluation demonstrated that the proposed methods were effective, achieving a macro-averaged F1 of 56.41% (rank 2nd) in English subtask A, 53.60 % (rank 3rd) in English subtask B and 67.70% (rank 3rd) in Arabic task, respectively.

1 Introduction

In recent years, community question answering (CQA) systems are becoming more and more popular on the Internet. By using CQA system, a user can post his/her question on CQA portal and receive answers from other users. All users can post questions and answers on CQA portal freely. Although it makes CQA users to get answers easily, the answer quality evaluation becomes a challenge for questions with multiple answers. To reduce the inconvenient in going through plenty of candidate answers, it makes sense to evaluate the quality of answers and select high-quality answers automatically for CQA systems. As a consequently, the task of answer quality evaluation and answer selection in CQA have attracted more and more attention in recent years (Arai and Handayani, 2013; Shah and Pomerantz, 2010; Agichtein et al., 2008).

The Answer Selection in CQA challenge was opened as one new task in SemEval-2015: SemEval-2015 Task 3 (Màrquez et al., 2015). It created a venue and provided annotated datasets for researchers to compare their methods for answer selection in CQA. This challenge consisted of Subtask A and Subtask B. Subtask A required participant system to classify answers as *relevant, potentially useful* and *bad* for each question. Subtask B required participant system to decide whether the answer to a *YES_NO* question should be *Yes*, *No* or *Unsure* based on the answer list. Subtask A was offered for two languages: English and Arabic. Data for the two languages was in different data set format. In remainder of this paper, Subtask A in English is abbreviated to English subtask A, Subtask A in Arabic is abbreviated to Arabic task and Subtask B in English is abbreviated to English subtask B.

HITSZ-ICRC team participated in English subtask A, English subtask B and Arabic task. This paper describes the ensemble learning method and hierarchical classification method proposed for each subtask in SemEval-2015 Task 3.

2 Methods for Answer Classification

Different classification methods were tried by previous researchers for answer evaluation, prediction and selection in CQA. Jeon et al. (2006) designed a framework using non-textual features, most of which were user profile features, to predict the document quality and tried the framework on CQA.

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 196–202,
Denver, Colorado, June 4-5, 2015.

Shah and Pomerantz (2010) used text, user information and answer rank features to evaluate and predict answer quality. Arai and Handayani (2013) tried non-textual features mainly include no-content features of text to train models to predict answer quality in CQA. For SemEval-2015 Task 3, we proposed ensemble learning method and hierarchical classification method to classify answers for each task.

2.1 English subtask A

English subtask A required participant system to classify each answer of test questions as definitely relevant (*good*), potentially useful (*potential)* or bad (*bad*, *dialog*, *non-English* and *other*).

Features employed to train classifiers for English subtask A include:

Word length features: length of the max length word, average word length.

Word number features: word number, capital word number, polite word number, word "yes" number, word "no" number, word "thank" number.

Punctuation features: question mark number, exclamation mark number.

Sentence features: average sentence length, sentence number.

Part-of-speech features: noun word number and ratio, verb word number and ratio, pronoun word number and ratio, WH word number and ratio.

Name entity feature: number of name entity.

Content tag features: number of web link and number of image link contained in content.

The 7 groups features in the upper list were extracted separately on questions and answers.

Answer position in Answer list: whether the answer is first, whether the answer is last.

User id features: whether user id of answer is the question user id, whether the user id of previous answer is question user id, whether the user id of next answer is question user id.

Answer and question correlative features: number and ratio of same n-gram terms between answer and question, cosine similarity between answer body and question body, KL distance between answer body and question body.

Class tag features: QCATEGORY tag of question, QTYPE tag of question.

Frequent n-gram term features: frequent unigram terms, bigram terms and trigram terms.

Two methods were proposed to classify answers for English subtask A: (1) two-level hierarchical classification: classifying answers as *good_potential* and *bad_dialog* in the first level; classifying *good_potential* answers as *good* and *potential*, classifying *bad_dialog* answers as *bad* and *dialog* separately in the second level; (2) ensemble learning: training and choosing top N best classifiers based on cross validation on training data, then using the N classifiers to vote final result.

2.2 English subtask B

The English subtask B required participant system to give "*Yes*", "*No*" or "*Unsure*" answer directly to a *YES_NO* question based on its candidate answers.

Evidence to answer *YES_NO* question is the *yes*/*no* opinion of each *good* answer in answer list. *YES_NO* question answering can be split into three steps: first, finding out *good* answers from candidate answers; second, classifying each *good* answer into *yes*, *no* or *unsure* based on its opinion; third, summarizing final answer for *YES_NO* question according to opinions of all *good* answers.

Given a *YES_NO* question, recognizing *good* answers can be achieved with the classifiers trained in English subtask A; final answer is predicted based on the comparison between the number of *yes* class answers and the number of *no* class answers in answer list of the question. So the remaining task for *YES_NO* question answering is *good* answer classification according to the opinion.

Two methods were proposed for answer opinion classification: (1) piping the best performance classifier for answer selection and the best classifier for answer opinion classification; (2) classifying answers of *YES_NO* question into 5 classes with single classifier: *yes*, *no*, *unsure*, *bad* and *dialogue*.

Feature extraction for English subtask A was same as English subtask A. Features employed were selected according to gain ratio. We proposed ensemble learning method for the answer classification in English subtask B.

2.3 Arabic task

Dataset for Arabic task is in Arabic. The task required participant system to classify answers of question into definitely relevant (*direct*), potentially useful (*related)* and bad (*irrelevant*).

Features extracted for Arabic task are similar to English subtask A. But some features were not extracted for Arabic task, such as "answer position"

was ineffective for Arabic task; "WH word number" cannot be extracted on Arabic data. To get more effective features, the dataset for Arabic task was translated to English by Google Translate[1], and feature extraction was done on both original Arabic data and English data translated from original Arabic data.

Features extracted for answer classification in Arabic task include:

Word length features: length of the max length word, average word length.

Word number feature: number of words.

Punctuation features: question mark number, exclamation mark number.

Sentence features: average sentence length, sentence number.

The features in the upper list were extracted separately on answers and questions.

Answer and question correlative features: number and ratio of same n-gram terms, cosine similarity between answer and question body, KL distance between answer and question body.

Name entity feature: number of name entity in answer.

Frequent n-gram term features: frequent unigram, bigram terms and trigram terms in Arabic data and English data.

Features were extracted only on translated English data in the following 2 groups:

Word number features in English: all capital word number, polite word number, word "yes" number, word "no" number.

Part-of-speech features: noun word number and ratio, verb word number and ratio, pronoun word number and ratio, WH word number and ratio.

Methods proposed for Arabic task include: (1) two-level hierarchical classification method: classifying answers as *irrelevant* and *not irrelevant* in the first level and classifying *not irrelevant* answers as *direct* and *related* in the second level; (2) ensemble learning method: training and choosing top N best classifiers and using the results of those classifiers to vote final result.

3 Data Sets

Data sets used for classifiers training includes the training and development data provided. No external data was used for classifiers training.

For English task, CQA-QL corpus (Màrquez et al., 2015) was provided. This corpus was gotten from the Qatar Living Forum[2] and was filtered and annotated manually. Questions in the corpus were labeled into *GENERAL* and *YES_NO* class in *QTYPE* dimension, and *yes*, *no*, *unsure* and *Not Applicable* class in *QGOLD_YN* dimension. Answers were labeled into *Good*, *Potential*, *Bad*, *Dialogue*, *Not English* and *Other* class in *CGOLD* dimension, and *Yes*, *No*, *Unsure* and *Not Applicable* class in *CGOLD_YN* dimension.

For Arabic task, Fatwa corpus (Màrquez et al., 2015) was provided, which was manually processed and annotated on source data from the Fatwa website[3]. Answers in this corpus were labeled into *direct*, *related*, and *irrelevant* class. The *irrelevant* class answers for each question were random selected from answers of other questions.

4 Results Evaluation

Some toolkits were employed to extract features and train classifiers. NLTK (Bird et al., 2009) was used to extract features, include part-of-speech of question and answer, frequent n-gram terms, cosine similarity and so on. WEKA (Hall et al., 2009) toolkit was used to do feature selection and classifier training and choosing. LIBSVM (Chang and Lin, 2011) and LIBLINEAR (Fan et al., 2008) were used to train SVM classifier. Scikit-learn toolkit (Pedregosa et al., 2011) was used to train classifiers.

We submitted 3 formal results for each subtask including English subtask A, English subtask B and Arabic task following task result submission requests: 1 primary result as team official result, 2 contrastive results to compare effects of different methods.

4.1 Measures

The official metric to evaluate results is the macro-averaged *F1-score* (Màrquez et al., 2015), which is calculated as:

$$macro - F1 = \frac{\sum_{i=1}^{NumC} F1_i}{NumC} \quad (1)$$

where *NumC* is the number of class in test set, $F1_i$ is the *F1* value for class *i* in test set. *F1* value is calculated as:

[1] http://translate.google.com

[2] http://www.qatarliving.com/forum

[3] http://fatwa.islamweb.net

$$F1 = \frac{2 \times P \times R}{P + R} \quad (2)$$

where P and R is the precision and recall of test results for a class in test set.

The total *accuracy* for test result is used as secondary metric for results comparison, which is calculated as:

$$Accuracy = \frac{totalRighNum}{totalTestCaseNum} \quad (3)$$

4.2 Results of English subtask A

Official evaluation on English subtask A was different to other task. In CQA-QL corpus, all answers were labeled in fine-grained labels which include 6 classes: *good*, *bad*, *potential*, *dialogue*, "*not English*" and *other*. But in official evaluation, the *macro-F1* score was calculated based on the coarse-grained labels which include 3 classes: *good*, *bad*, *potential*. The class *dialogue*, "*not English*" and *other* were merged with class *bad*.

We considered English subtask A as a 5-class (*good*, *potential*, *bad*, *dialogue*, and "*not English*") classification problem. The answers in "not English" class were firstly recognized by toolkit Language Detection (Shuyo, 2010). Other answers were classified with methods we proposed.

The evaluation results for English subtask A submissions are shown in table 1.

Submission	Macro F1	Accuracy
primary	56.41	68.67
contrastive1	**56.44**	**69.43**
contrastive2	55.22	67.91

Table 1. Macro F1 and accuracy of English subtask A.

The primary submission was gotten by two-level hierarchical classification method: in the first level, answers were classified into *good_potential* and *bad_dialogue*. In the second level, *good_potential* answers and *bad_dialogue* answers were classified separately: *good_potential* answers were classified into *good* and *potential*, *bad_dialogue* answers were classified into *bad* and *dialogue*. The classifiers used here were SVM which were trained using toolkit LIBLINEAR.

In contrastive1 submission, two-level hierarchical classification method was used, and a special ensemble learning method was designed for *potential* answers classifying. The *potential* class answers were classified using ensemble learning method in the first level. The other 3 classes answers were classified in the second level. The ensemble learning method for *potential* answers classification using 5 binary classifiers: 3 *good-potential* classifiers trained using different training data; 1 *bad-potential* classifier and 1 *dialogue-potential* classifier. The training data for *good-potential* classifiers was gotten by random splitting *good* answers into 3 parts. Classifiers used for the contrastive1 submission were SVM trained with toolkit LIBLINEAR.

Steps for getting the contrastive2 submission were similar to the primary submission. The difference was that the first level classifier was trained using Random Forest algorithm (Breiman, 2001). The training data *good-potential* classifier was resampled to balance the instance distribution between *good* and *potential* class.

Features employed for English subtask A includes 4044 features: the top 4000 frequent n-gram terms and the top 44 maximum gain ratio features of all the features described in section 2.1 except the "Frequent n-gram term features".

4.3 Results of English subtask B

Three submissions were submitted for English subtask B including primary submission, contrastive1 submission and contrastive2 submission. The evaluation results are presented in table 2.

Submission	Macro F1	Accuracy
primary	**53.60**	**64.00**
contrastive1	42.50	60.00
contrastive2	42.40	60.00

Table 2. Macro F1 and accuracy of English subtask B.

For the primary submission, answers in *YES_NO* question answer list were classified into 5 classes. Steps to classify answers in *CGOLD_YN* dimension were: first, a rule based method was used to classify answers; second, ensemble learning method was used to classify the answers that cannot be classified by rule based method. Classifiers used in ensemble learning method include: SMO (sequential minimal optimization algorithm for SVM) (Keerthi et al., 2001), Random Forest, DMNBtext (Discriminative Multinomial Naïve Bayes) (Su et al., 2008), Logistic Regression (Le Cessie and Van Houwelingen, 1992) and RBFNetwork (normalized Gaussian radial basis function network). Those classifiers were the top 5 best of all classifiers have been tried based on 10 folds cross valida-

tion on training data. Features employed for the primary submission include 187 features, which were the top 187 maximum gain ratio features of the 4400 features used in English task A.

The contrastive1 submission and contrastive2 submission were based on the *good* answers in English subtask A primary submission. Only *good* answers of *YES_NO* question in subtask A primary submission were classified in *CGOLD_YN* dimension. *Good* answers of *YES_NO* question were classified into: *yes*, *no* and *unsure.*

For the contrastive1 submission, *good* class answers were classified with ensemble learning method. Classifiers used for the ensemble learning method included the top 5 best classifiers for answer classification in *CGOLD_YN* dimension: SMO, Random Forest, DMNBtext, Logistic Regression and LMT (logistic model tree) (Sumner et al., 2005).

For the contrastive2 submission, only classifier LMT, which was the best classifier of all classifiers tried based on 10 folds cross validation results on training data, was used to classify *good* answers.

Features employed for the contrastive1 and contrastive2 submission include 110 features, which were the top 110 maximum gain ratio features of the 4400 features used in English task A.

4.4 Results of Arabic task

Answers were classified into 3 classes in Arabic task: *direct*, *related*, and *irrelevant*. Evaluation results for Arabic task are presented in table 3.

The primary submission was gotten by ensemble learning method using 3 classifiers. The classifiers were top 3 classifiers chosen based on 10 folds cross validation results on training data: SMO, REPTree (decision/regression tree) and J48graft (grafted C4.5 decision tree) (Webb, 1999).

Submission	Macro F1	Accuracy
primary	67.70	74.53
contrastive1	**68.36**	**73.93**
contrastive2	67.98	73.23

Table 3. Macro F1 and accuracy of Arabic task.

The contrastive1 submission was gotten by two-level hierarchical classification method: in the first level, answers were classified into *irrelevant* and *not irrelevant*; in the second level, *not irrelevant* answers were classified into *direct* and *related*. All classifiers were trained using SMO algorithm.

The contrastive2 submission was gotten only by SMO classifier. The SMO classifier was trained as multi-class classifier to classify answers into *direct*, *related* and *irrelevant*.

Features employed for Arabic task include 5049 features: the top 5000 frequent n-gram terms and the top 49 maximum gain ratio features of all the features described in section 2.3 except "Frequent n-gram term features".

5 Discussion

In English subtask A, performance of the submission contrastive1, the hierarchical classification method result, was better than other submissions. The performance of hierarchical classification method was also better than other submission in Arabic task. This shows that the hierarchical classification method is effective for answer selection task.

The performances on different class varied from each other remarkable for English subtask A and Arabic task as shown in table 4. It is difficult to distinguish the potentially useful class answers for all classification methods that have been tried. Analysis on feature extraction showed that, most features were extracted to judge whether the answer was *good* or *bad*, but few features were extracted to judge whether the answer was potentially useful.

Submission	Class	P	R	F1
English subtask A contrastive1	*Good*	78.02	79.74	78.87
	Bad	80.6	66.01	72.58
	Pot.	14.04	24.55	17.86
English subtask B primary	*Yes*	80	80	80
	No	28.57	50	36.36
	Unsure	66.67	33.33	44.44
Arabic task contrastive1	*direct*	62.4	74.88	68.08
	Irrel.	85.14	83.33	84.23
	related	57.07	49.1	52.78

Table 4. Detailed evaluation results (P, R and F1) of the best performance result for each task.

In English subtask B, performance on primary submission, which was result of one-step classification method on all answers of YES_NO question, was much better than other submissions which were results of two-step classification method. The results showed that cascade error of piping classifiers for answer classification in *CGOLD* and answer classification in *CGOLD_YN* had great im-

pact on final answer accuracy for *YES_NO* question. The one-step classification method can avoid the cascade error for *Yes_NO* questions answering.

We compared performance of SVM classifier using bag-of-word features, non-bag-of-word features and all features for English subtask A, subtask B and Arabic task on *macro-F1* scores. The results are shown in table 5.

Task	*bow*	*non_bow*	*bow+non_bow*
Subtask A	0.39	0.48	0.50
Subtask B	0.42	0.64	0.68
Arabic Task	0.36	0.35	0.42

Table 5. Macro F1 of SVM classifier using bag-of-word features, non-bag-of-word features and all features.

Feature set bag-of-words (*bow*) includes **Frequent n-gram term features** described in section 2.1 and 2.3. Feature set non-bag-of-words (*non_bow*) includes other features described in section 2.1 and 2.3 which were specially designed for answer selection task. Set *bow+non_bow* includes all features in set *bow* and *non_bow*.

The performance of the classifier using *bow+non_bow* features is better than using the other two sets features in isolation, which means *bow* set features and *non_bow* set features are effective to improve performance of answer classifier if used both. The contribution of different sets is different on different tasks. Performance of *non_bow* (44 features for English data and 49 features for Arabic data) is better than *bow* (4000 for English and 5000 for Arabic) on Answer Selection task. It shows the features specially extracted for answer selection are more effective. But performance of *non_bow* (22 features) is worse than *bow* (165 features) on *YES_NO* questions answering. The reason is that the *non_bow* features are not designed for opinion recognition. It shows that designing special features for opinion recognition for task B is necessary.

6 Conclusions and Future Work

In this paper, we presented multi-classifier ensemble method and hierarchical classification method proposed for each subtask in SemEval-2015 Task 3. Experimental results demonstrated that the proposed classification methods were effective in both English and Arabic subtasks.

In the next stage, syntax feature and deep semantic feature will be exploited to further improve the performance of our approaches. Besides, more effective features for *potential* answers classification will also be explored.

Acknowledgments

The authors thank Daniel Cer and all the anonymous reviewers for their insightful comments for this paper.

This work was supported in part by the National Natural Science Foundation of China (61272383, 61173075 and 61203378), the Strategic Emerging Industry Development Special Funds of Shenzhen (ZDSY20120613125401420 and JCYJ20120613151940045) and the Key Basic Research Foundation of Shenzhen (JC201005260118A).

References

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1-27.

Chirag Shah and Jefferey Pomerantz. 2010. Evaluating and Predicting Answer Quality in Community QA. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 411-418, Geneva, Switzerland, 19-23 July.

Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008. Finding High-quality Content in Social Media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, 183-194, Palo Alto, California, USA, 11-12 February.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research*, 12:2825-2830.

Geoffrey I Webb. 1999. Decision tree grafting from the all-tests-but-one partition. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 2:702-707, San Francisco, California, USA.

Jiang Su, Harry Zhang, Charles X Ling, and Stan Matwin. 2008. Discriminative parameter learning for Bayesian networks. In *Proceedings of the 25th international conference on Machine learning*, 1016-1023, Helsinki, Finland.

Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. 2006. A Framework to Predict the Quality of Answers with Non-textual Features. In *Proceedings*

of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 228-235, Seattle, Washington, USA, 6-11 August.

Kohei Arai and Anik Nur Handayani. 2013. Predicting quality of answer in collaborative Q/A community. *Society and culture*, 2(3):21-25.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5-32.

Lluís Màrquez, James Glass, Walid Magdy, Alessandro Moschitti, Preslav Nakov, 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)*, Denver, Colorado, USA.

Marc Sumner, Eibe Frank, and Mark Hall. 2005. Speeding up logistic model tree induction. In *Knowledge Discovery in Databases: PKDD 2005*, 3721:675-683.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10-18.

Nakatani Shuyo. 2010. *Language Detection Library for Java*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *The Journal of Machine Learning Research*, 9:1871-1874.

Sathiya Sathiya Keerthi, Shirish Krishnaj Shevade, Chiru Bhattacharyya, and K. R. K. Murthy. 2001. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation*, 13(3):637-649.

Saskia Le Cessie and Johannes C Van Houwelingen. 1992. Ridge estimators in logistic regression. *Applied statistics*, 191-201.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python*. O'Reilly Media, Inc. .