ECNU: Using Multiple Sources of CQA-based Information for Answer Selection and YES/NO Response Inference

Liang Yi, Jianxiang Wang, Man Lan*
Shanghai Key Laboratory of Multidimensional Information Processing
Department of Computer Science and Technology,
East China Normal University, Shanghai 200241, China
{51121201035, 51141201062}@ecnu.cn; mlan@cs.ecnu.edu.cn*

Abstract

This paper reports our submissions to community question answering task in SemEval-2015, which consists of two subtasks: (1) predict the quality of answers to given question as *good*, *bad*, or *potentially relevant* and (2) identify *yes*, *no* or *unsure* response to a given YES/NO question based on the *good* answers identified by subtask 1. For both subtasks, we adopted supervised classification method and examined the effects of heterogeneous features generated from community question answering data, such as bag-of-words, string matching, semantic similarity, answerer information, answer-specific features, question-specific features, etc. Our submitted primary systems ranked the forth and the second for the two subtasks of English data respectively.

1 Introduction

Community Question Answering (CQA) systems such as *Yahoo!Answers* rely on users to provide answers (i.e., user generated content) for questions posted. Generally such systems are quite open and the answers provided by users are not always of high quality. For example, a bad answer may present irrelevant opinions or issues, contain only URL links without direct answer, or even be written informally. Therefore, in order to achieve high-quality user experience and maintain high levels of adherence, it is critical to present high-quality answers and provide direct responses for users.

The CQA task in SemEval-2015 (Màrquez et al., 2015) provides such a universal platform for researchers to make a comparison between different approaches. This task consists of two subtasks: (1) subtask A is to classify the quality of answers as *good*, *potential* or *bad*, which also refers to the task of answer quality prediction (Jeon et al., 2006; Agichtein et al., 2008); (2) subtask B is to infer the global answer of a YES/NO question to be *yes*, *no* or *unsure* based on individual *good* answers.

Most of the previous research on answer quality prediction has focused on extracting various features to employ ranking or classification methods (Surdeanu et al., 2011; Shah and Pomerantz, 2010), such as textual features (Agichtein et al., 2008; Blooma et al., 2010) including the length of an answer, overlapped words between a question-answer (QA) pair, etc. Another kind of widely used feature is extracted from answerer profile information (Shah and Pomerantz, 2010), such as the number of best answers, the achieved levels and the earned points. However, such information is not often available in real world. Moreover, a recent study (Toba et al., 2014) has taken question type into consideration to make the answers quality prediction.

In this paper, we built two classification systems for the two tasks respectively. For Task A, we extracted six types of features from multiple sources of CQA-based information to predict the answer quality, such as answer-, question-, answerer-specific information, surface word similarity and semantic similarity between question-answer pair, ect. For Task B, the global answer of a YES/NO question is summarized just from the individual *good* answers identified by Task A. Specifically, we first built a classifier to predict *Yes*/*No*/*Unsure* labels for each

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 236–241,
Denver, Colorado, June 4-5, 2015.

predicted *good* answer, then we performed a majority voting to summarize the global answer for each question.

The rest of this paper is structured as follows. Section 2 describes our systems, including features, algorithms, etc. Section 3 shows experiments on training data and results on test data. Finally, conclusions and future work are given in Section 4.

2 Our Systems

For both tasks we adopted supervised classification methods and extracted various features from multiple sources to predict answer quality and infer YES/NO response.

2.1 Data Extraction

English data is extracted from Qatar Living Forum[1] and provided with XML-format. Each data file consists of a list of question tags, where each question is followed by a list of answer tags to this question. Each question or answer has a subject, a body, and a list of attributes from which we can extract significant features. For example, a question has attributes of question category (overall 27 categories, e.g., Education, Cars, etc.), identifier of asker, question type (GENERAL or YES/NO) and an answer also has answerer identifier.

To obtain complete contents of a question or an answer, we merged the contents extracted from subject and body. Exceptionally, if subject is substring of body or subject of an answer starts with "RE:", we just extracted the contents from body.

Moreover, to reduce the influence of *Not English* answers to the subsequent classification, we filtered out the *Not English* answers from data. To discover such answers we found out unusual words for each answer by comparing word set of this answer with an English vocabulary with $235,887$ words from NLTK[2] *words* corpus, if the number of unusual words is over 10 and the ratio over answer length is above 60% we then regarded it as *Not English*.

2.2 Pre-processing

After data extraction we performed the following preprocessing operations. Firstly, HTML character encodings are substituted by the actual characters (e.g., "&" is converted into whitespace). Then HTML tags, URLs, emoticons, ending signatures and repeating punctuation are removed from data. After that, we collected a slang list from Internet and replaced the informal words with formal words (e.g., "*u r*" is converted into "*you are*"). For the processed data, we performed tokenization and POS tagging using Penn Treebank tokenizer and POS tagger in NLTK. The words are lemmatized using WordNet-based lemmatizer implemented in NLTK.

2.3 Features of Task A

We extracted six types of features from multiple sources of CQA-based information, i.e., bag-of-words (BoW) and answer-specific features (AS) from answer, string matching (SM) and semantic similarity (SS) from QA pair, answerer information features (AI) from answerer profile, question-specific features (QS) from question.

2.3.1 Bag-of-Words for Answer (BoW)

We collected words from training and development answer set and adopted binary BoW representation. To reduce the problem of data sparse, we selected the words with frequency higher than four, resulting in $5,730$ words.

2.3.2 Answer-Specific Features (AS)

For each question, we extracted three answer-specific features. The first is answer length, which is computed at three levels, i.e., word, sentence and paragraph. We used L_1 normalization on the global answer set. To gain insight on the effect of answer length for each individual question, we also designed a length ratio feature to record the ratio of the length of each answer to the maximal answer length for the same question.

A good answer is generally supposed to answer a question explicitly instead of starting a new question or suggesting other consulting approaches. Therefore, the second binary feature is to represent whether an answer contains a question mark or not. In addition, we manually collected eight words and phrases from training set, which contains the meaning of suggestion (i.e., "*suggest*", "*recommend*", "*advise*", "*try*", "*call*", "*you may*", "*may be*", "*you could*"). Thus the third binary feature is to

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

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

represent if there is at least one of above suggestion words in a given answer.

2.3.3 String Matching between QA (SM)

The above two types of features are both extracted from answer regardless of the question asked. However, the string matching features are to consider the overlapped words from a given QA pair.

Word: This feature group records the proportions of co-occurred words between a QA pair, which are calculated using six measures: $|A \cap B|/|A|$, $|A \cap B|/|B|$, $|A-B|/|A|$, $|B-A|/|B|$, $|A \cap B|/|A \cup B|$, $2*|A \cap B|/(|A|+|B|)$, where $|A|$ and $|B|$ denote the number of non-repeated words of question A and answer B. However, the same word appearing in different context could vary in word forms and normalizing words may obtain more accurate overlapped proportions, so we computed each measure at three word forms: original, lemmatized and stem form.

POS: This POS feature is similar to the above word feature. We use three measures: $|A \cap B|/|A|$, $|A \cap B|/|B|$, $|A \cap B|/|A \cup B|$ to compute overlapped proportion of POS tags for nouns, verbs, adjectives and adverbs.

n-gram: Unlike the above two features measuring the overlap of single words or POS without considering multiple continuous words, the *n*-gram feature is to calculate the Jaccard similarity of overlapped n-grams between each QA pair. The *n*-grams are obtained at word level ($n = 2, 3$) and character level ($n = 2, 3, 4$). In addition, the *n*-grams at word level are obtained from original form and lemmatized form respectively.

Longest Common Sequence (LCS): The LCS feature is to measure the LCS similarity for a QA pair on the original and lemmatized form. It is calculated as the length of the LCS between each QA pair at word level divided by the length of question.

2.3.4 Semantic Similarity between QA (SS)

The previous string matching feature only considers the overlapped surface words or substrings in a QA pair and it may not capture the semantic information between a QA pair. Therefore, we presented the following semantic similarity features, which are borrowed from previous work.

Determining semantic similarity of sentences commonly uses measures of semantic similarity between individual words. We used knowledge-based and corpus-based word similarity features. The knowledge-based similarity estimation relies on a semantic network of words such as WordNet. In this work, we employed four WordNet-based word similarity metrics: *Path* (Banea et al., 2012), *WUP* (Wu and Palmer, 1994), *LCH* (Leacock and Chodorow, 1998) and *Lin* (Lin, 1998) similarity. Following (Zhu and Man, 2013), the best alignment strategy and the aggregation strategy are employed to propagate the word similarity to the text similarity. Moreover, Latent Semantic analysis (LSA) (Landauer et al., 1997) is a widely used corpus-based measure when evaluating textual similarity. We used the vector space sentence similarity proposed by (Šarić et al., 2012), which represents each sentence as a single distributional vector by summing up the LSA vector of each word in the sentence. In this work, two corpora are used to compute the LSA vector of words: New York Times Annotated Corpus (NYT) and Wikipedia.

Besides, following (Zhao et al., 2014), we adopted the weighted textual matrix factorization (WTMF) (Guo and Diab, 2012) to model the semantics representations of sentences and then employed the new representations to calculate the semantic similarity between QA pairs using Cosine, Manhattan, Euclidean, Person, Spearmanr, Kendalltau measures respectively.

2.3.5 Answerer Information (AI)

Previous work (Zhou et al., 2012) showed that information about answerer has great impact on answer ranking in CQA. Inspired by this work, we designed two answerer-specific features to represent answerer level and answerer expert domain information. To calculate the answerer level feature, we used the number of answers and the percentage of *good* answers for each answerer. For expert domain feature, we employed the question categories where the answerer is an expert. Specifically, for each answerer, let G be the number of *good* answers the answerer responses and G_i be the number of *good* answers to the *i*-th question category (i<=27). Then we used G_i/G to measure the answerer's expert domain. Besides, for each of the 27 question categories (e.g., Education, Cars), we recorded the maximal value M_i over all values of G_i from each answerer and then

calculated the G_i/M_i score to measure expert level of an answerer in current domain among all answerers. Totally, we adopted 54 features to indicate expert domain for each answerer.

2.3.6 Question-Specific Features (QS)

Since the domain of questions may also affect the performance of answer selection, we considered to use 27 binary features to indicate the question category. In addition, we manually collected 9 question words (i.e., *where*, *what*, *when*, *which*, *who*, *whom*, *whose*, *why* and *how*) and used 9 binary features to indicate if one of these question words occurs in the question.

2.4 Features of Task B

To address task B, we performed two steps. Firstly, we extracted features from *good* answers identified from task A and trained a classifier to predict the *Yes*, *No* or *Unsure* label for each *good* answer. Secondly, for each given YES/NO question, we counted the answer labels of *Yes*, *No* or *Unsure* and used majority voting to obtain the global answer.

We used three types of features for this task, which are all extracted from answer: (1) Bag-of-Words from answer (BoW), the same as in Task A; (2) Semantic Word2Vec (W2V): this feature indicates a vector representation of answer. We used word2vec tool[3] to get word vectors with dimension $d = 300$ and then summed up all the word vectors to obtain the answer vector. (3) Yes/No Word List (YN): we manually collected 50 affirmative words and 45 negation words by starting from several seed words (e.g., "*yes*", "*sure*", "*definitely*", "*no*", "*seldom*", "*never*", etc) and then expanding the list using snowball with the aid of WordNet synset. Besides, several phrases are manually added in the list (e.g., "*beyond a doubt*", "*beyond question*", "*not at all*", "*only just*", etc). We utilized 2 binary features to indicate whether an answer contains at least one of these affirmative and negation words or not.

2.5 Classification Algorithms

We explored several widely-used supervised classification algorithms including Support Vector Machine (SVM), Random Forest (RF), and Gradient Boosting(GB), which are implemented in scikit-learn toolkit (Pedregosa et al., 2011).

[3]https://code.google.com/p/word2vec/

Algorithm	macro-F_1(Task A)	macro-F_1(Task B)
SVM (linear)	**54.25**	**58.60**
SVM (rbf)	29.44	25.05
GB	49.70	39.05
RF	45.40	27.14

Table 1: Results on training data for different algorithms.

2.6 Evaluation Measures

The official evaluation measures for both tasks is macro-averaged F_1. For Task A the official score is calculated on three labels: *Good*, *Bad*, *Potential* (where *Bad* includes *Dialogue*, *Not English* and *Other*).

3 Experiments and Results

3.1 English Data Set

The English training and development set contain $2,900$ questions with $18,186$ answers and the test set contains 329 questions with $1,976$ answers, consisting of around 50% *good*, 40% *bad* and 10% *potential* answers. The YES/NO questions are about 10% of all the questions, which indicates that the data for Task B is much less than Task A.

For both tasks we used training set with $2,600$ questions to build classifiers and validated the performance on development set with 300 questions for algorithms comparison and features choosing.

3.2 Algorithm Choosing Experiments

We performed algorithm choosing experiments using all designed features. All the parameters of algorithms are set to be default values from scikit-learn (Pedregosa et al., 2011). Table 1 lists the preliminary algorithm comparison experimental results. We found SVM with linear kernel outperforms other algorithm choices for both tasks. Moreover, we tuned the trade-off parameter *c* of SVM and when set *c* to 0.8 we obtained a better score 54.78% and 58.82% for Task A and B respectively. Therefore, in the following experiments on training and test data, we set the algorithm to SVM with linear kernel.

3.3 Feature Comparison Experiments

We performed a series of experiments for both tasks to explore the effects of various feature types using

SVM (linear). In Task B we always chose the predicted *good* answers from the system with the best macro-F_1 in Task A. Table 2 shows the results of different feature combinations where for each time we selected and added one best feature type. From this table we found the following interesting observations.

	BoW	AS	SM	SS	AI	QS	macro-F_1(%)
Task A	+						48.91
	+	+					49.73(+0.82)
	+	+			+		51.85(+2.12)
	+	+	+		+		52.03(+0.18)
	+	+	+	+	+		53.22(+1.19)
	+	+	+	+	+	+	**54.25**(+1.03)

	BoW	W2V	YN	macro-F_1(%)
Task B	+			47.82
	+	+		49.54(+1.72)
	+	+	+	**58.60**(+9.06)

Table 2: Results of feature combinations for Task A and B, the numbers in the bracket are the performance increments compared with previous result.

First, for both tasks the most effective feature type is bag-of-words from answer and this feature alone achieves 48.91% for Task A and 47.82% for Task B, which both outperforms the baseline system provided by organizers respectively. The baseline of Task A which predicts all answers as *good* just achieves 22.36% and for Task B it achieves 25.0% which predict all answers as *yes*. Moreover, in Task A the performance of other five feature types alone is far lower than bag-of-words, ranging from 23% to 38% approximately.

Second, for Task A, when combining all the features together the system achieves the best performance, which indicates that all types of features make contribution more or less. Specially, among the six types of features, answerer information and semantic similarity between QA pairs make more contribution than others. This indicates that answerer profile information is important, which is consistent with the findings in (Zhou et al., 2012). Besides, the semantic similarity captures deep relationship between Q-A pair than the surface word, which is helpful for performance improvement. In Task B, we also observed the similar findings, i.e., the system using all types of features achieves the best performance. Moreover, the YES/NO word list feature makes great contribution to the performance improvement. This is consistent with our expectation. Besides, although in this work the word vector feature improves the performance, this improvements is not as much as our expectation. The possible reason may be the simple way of using the vector by only summing up.

3.4 Results on Test Data

According to the above experiments on training data, we configured one primary and two contrastive systems for both tasks. The only difference between these systems lies in the features and parameters in SVM. Table 3 lists the configuration of three systems and their corresponding results on test data. Besides, we also list the top three results officially released by organizers.

Systems	Task A			Task B		
	features	para.	result	features	para.	result
primary	all	c=0.8	**53.47**(9)	all	c=0.8	**55.8**(3)
contrastive1	all	c=1.0	52.55(10)	all	c=1.0	50.6(4)
contrastive2	all-SS	c=0.8	52.27(11)	all-W2V	c=0.8	53.9(6)

Top Systems	Task A Result	Task B Result
rank 1st	**57.19**	**63.7**
rank 2nd	56.41	55.8
rank 3rd	53.74	53.6

Table 3: Configurations and results of our three submitted systems and top three results, the numbers in bracket are the official ranking out of all submitted systems.

Our primary system ranked the 4th out of 12 participants in Task A and the 2nd out of 7 participants in Task B. For both tasks the performance of the primary system is higher than the two contrastive systems, which is consistent with the results on training data.

4 Conclusion

We build two supervised classification systems for answer selection and YES/NO response inference in CQA. Specially, we extract heterogeneous features from various information sources, i.e., answer, question, answer-question pair and answerer. Our experiments reveal that our designed features are all effective and when we combine all types of features together the system achieves the best performance.

Although multiple features extracted from CQA, the way of using these features are quite simple. Besides, due to the huge number of bag-of-word feature, the effects of other specific features are impaired. For future work, we may explore other underlying useful features and the advanced way of integrating these features to further improve the performance, such as the fine-grained semantic relationship between question and answer, etc.

Acknowledgments

This research is supported by grants from Science and Technology Commission of Shanghai Municipality under research grant no. (14DZ2260800 and 15ZR1410700) and Shanghai Collaborative Innovation Center of Trustworthy Software for Internet of Things (ZF1213).

References

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*, pages 183–194.

Carmen Banea, Samer Hassan, Michael Mohler, and Rada Mihalcea. 2012. Unt: A supervised synergistic approach to semantic text similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 635–642.

Mohan John Blooma, Alton Yeow-Kuan Chua, and Dion Hoe-Lian Goh. 2010. Selection of the best answer in cqa services. In *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, pages 534–539.

Weiwei Guo and Mona Diab. 2012. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 864–872.

Jiwoon Jeon, William 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*, pages 228–235.

Thomas K Landauer, Darrell Laham, Bob Rehder, and Missy E Schreiner. 1997. How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans. In *Proceedings of the 19th annual meeting of the Cognitive Science Society*, pages 412–417.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304.

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 *SemEval 2015*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 441–448.

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*, pages 411–418.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2):351–383.

Hapnes Toba, Zhao-Yan Ming, Mirna Adriani, and Tat-Seng Chua. 2014. Discovering high quality answers in community question answering archives using a hierarchy of classifiers. *Information Sciences*, 261:101–115.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138.

Jiang Zhao, Tian Tian Zhu, and Man Lan. 2014. Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. *SemEval 2014*, page 271.

Zhi-Min Zhou, Man Lan, Zheng-Yu Niu, and Yue Lu. 2012. Exploiting user profile information for answer ranking in cqa. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 767–774.

Tian Tian Zhu and LAN Man. 2013. Ecnucs: Measuring short text semantic equivalence using multiple similarity measurements. *Atlanta, Georgia, USA*, page 124.