An Effective Category Classification Method Based on a Language Model for Question Category Recommendation on a cQA service

Kyoungman Bae and Youngjoong Ko
Department of Computer Engineering, Dong-A University
840, Hadan 2-dong, Saha-gu, Busan, 604-714, Republic of Korea
Kmbae0722@gmail.com, yjko@dau.ac.kr

ABSTRACT
Classifying user’s question into several topics helps respondents answering the question in a cQA service. The word weighting method must estimate the appropriate weight of a word to improve the category (or topic) classification. In this paper, we propose a novel effective word weighting method based on a language model for automatic category classification in the cQA service. We first calculate the occurrence probability of a word in each category by using a language model and then the final weight of each word is estimated by ratio of the occurrence probability of the word on a category to the occurrence probability of the word on the other categories. As a result, the proposed method significantly improves the performance of the category classification.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; I.2.7 [Natural Language Processing]: Text analysis

Keywords
Category (or Topic) Classification, cQA Service, Language Model, Word Weighting, Category (or Topic) Recommendation

1. INTRODUCTION
Effectively responding to other users’ questions is very important to build up a community-based Question Answering (cQA) service because a user cannot get any information about his/her question not having any answer. One solution to this problem is the automatic category (or topic) classification of user’s questions. The cQA service has used a category classification technique to recommend the candidate categories when a user enters his/her question into the cQA system. Since the cQA service provides a user with several candidate categories, the user can easily determine the categories of his/her question among the candidate categories. The categories of user’s question can help people search for relevant questions and create its answer. It makes a requirement for cQA services that they have to obtain higher precision than recall in their performance.

For this reason, recommending candidate categories have to be considered as an important function for the cQA service. For this, the efficient category classification method for input questions, which is the task of automatically assigning input questions into predefined categories (or topics), is needed. Question Classification (QC) is similar to our research. However, question classification is to assign one or more class labels to a given question written in natural language and the set of predefined classes are usually called question taxonomy or answer type taxonomy, such as “Location,” “Human,” and so on [1]. Since typical Question Answering (QA) systems automatically generate the answer, they use the question classification to detect the entity type of the answer that is the question taxonomy. For example, the class of question “Who is the tallest player in New York Yankees?” is a named entity of type “Human”. The focus of question classification in QA systems is to predict the type of the answer. On the other hand, the focus of question classification in cQA services is to classify and recommend the category or topic of the question because the answer in cQA services is directly generated by user. For example, the result of question classification in cQA services about the aforementioned question should be “Sports”. Question classification in cQA services is strongly associated with the category (or topic) classification.

In this paper, we propose a new category classification method for input questions in the cQA service that exploits a novel word weighting scheme based on a language model. In fact, category classification of the cQA service has a different aspect from traditional text classification; the input of category classification in the cQA service consists of a small size of text because it is a question, different from long texts in the traditional text classification. To solve this problem, we focus on the distribution of the word not in a question but in each category. The shorter input text, a question, has small number of content words to express its category and most of the content words occur just one time on its body. Therefore, we consider the role of word in each category as well as a text (term frequency) and whole corpus (inverse document frequency). For this, we first calculate the occurrence probability of a word in each category by using a language model. Then the final weight of each word is estimated by ratio of the occurrence probability of the word on a category to the occurrence probability of the word on the other categories.

We evaluated the proposed method by comparing the baseline models including TFIDF, OkapiBM25 and Language Model (LM). In our experiment, the proposed model achieved 94.2%, which is 12.3% higher performance than TFIDF. As a result, the proposed method improved the performance of category classification significantly.
This paper is organized as follows. In section 2, our word weighting method for category classification is described in detail. In section 3, we discuss the analysis of experimental results and discussion. Finally, we draw some conclusions in section 4.

2. Proposed Method

One of the major challenges for category (or topic) classification is to effectively estimate the weight of words in each category. Here, we deal with the problem of estimating and utilizing the distribution of words in categories. Especially, application areas just like a cQA service, which have an input text with smaller number of words, require more effective and discriminative word weighting method.

For this, we pay attention to a problem in traditional text classification. The traditional text classification has not used a category (or topic) distribution to estimate word weights; it has usually used document distribution (term frequency) and corpus distribution (inverse document frequency). We assume that a word that is deeply associated with a particular category but rarely related in the other categories has to be considered as a more important feature. Based on this assumption, we can estimate better quality of word weights to effectively classify the category of the question with the word.

Our strategy is to first estimate the occurrence probability of all words in each category by using a language model and then the final weight of each word is calculated by the ratio of the occurrence probability of a word on a category to the average of occurrence probabilities on the other categories. The essential point of our approach is that content words that mostly appear in a particular category will contribute more to the category classification.

2.1 Proposed category-based language model

Our model applied a language model to category classification for the cQA service by using categories as well as question-answer pairs. The traditional language model uses the distributions of a word in a document and a corpus in information retrieval (Jeon et al., 2008). In our cQA service, we concentrate on distribution of a word in a category, which can be calculated as follows:

\[ TP(w_i, c_k) = P_m(w_i | c_k) = \frac{c(w_i, c_k)}{|c_k|} \]  

(1)

Here, \( c_k \) is the current category, \( w_i \) is \( i \)th word in Question-Answering pair \((Q,A)\), \( |c_k| \) denotes the length of current category, \( c(w_i, c_k) \) denotes the frequency of word \( w_i \) in current category \( c_k \), and \( P_m(w_i | c_k) \) is the maximum likelihood estimate of word \( w_i \) in current category \( c_k \).

2.2 Calculating final word weights

\( P_m(w_i | c_k) \) considers the role of word in only the current category and it could have a problem to be revised. If a word frequently occurs in the current category but rarely occurs in the other categories, such word obviously has very strong classification ability. But it cannot be reflected in equation (1). Therefore, our new word weighting method for category classification estimates word weights in a category as follows:

\[ RateAvg(w_i, c_k) = \log \left( \frac{TP(w_i, c_k)}{TP(w_i, c_k') + CP(w_i, Cor)} \right) \]

(2)

We use the ratio of the occurrence probability of the word on a category to the average of occurrence probability on the other categories. \( n \) is the number of total categories and \( c_k' \) is the other categories. \( TP(w_i, c_k') \) is the averaged probability estimation of the word \( w_i \) in the other categories. If the occurrence probability of a word in a current category is higher than one in the other categories, the rate-based word weight in equation (2) becomes high; otherwise, it becomes low.

To avoid zero value in the divisor of equation (2), we add a very small value \( \lambda \) into the divisor. In addition, the distribution of the word in the corpus is added into numerator as well as denominator, and it works as an additive smoothing; corpus distribution has commonly been used in language models.

\[ RateAvgS(w_i, c_k) = \log \left( \frac{TP(w_i, c_k) + CP(w_i, Cor)}{TP(w_i, c_k') + CP(w_i, Cor)} \right) \]

(3)

\[ CP(w_i, Cor) = P_m(w_i | Cor) = \frac{c(w_i, Cor)}{|Cor|} \]

(4)

Here, \( Cor \) denotes a corpus. In the experiments, this word weighting method of equation (4) achieved better performance than that of equation (2). When the probability of the other categories is higher than that of the current category, word weights have negative values. We assume that words with negative weights are bad features for category classification. Thus, the proposed model does not use words with a negative weight by replacing their weights with zero scores.

3. Experimental Evaluation

3.1 Data Sets and Experimental Settings

To verify the proposed method, we collected a total of 14,882 question-answering \((Q,A)\) pairs that were constructed from the best \(Q&A\) group in Naver Kin. Among 14 categories on Naver Kin, we selected 10 categories \((Computer, Communication, Game, Travel, Shopping, Sport, etc.)\) for our experiments; 4 removed categories do not have any \(Q&A\) pairs yet or too small number of the \(Q&A\) pairs because they are created lately. The size of \(Q\)As in Naver Kin remarkably varies in the number of the content words. Note that most of \(Q\)As contains only about 10 content words.

For fair evaluation, we used 10-fold cross evaluation. As an evaluation measure for the performance of category classification, we computed the F1 measure.

\[ F1 = \frac{2PR}{P+R} \]

(6)

In addition, we used SVMs as the state-of-the-art classifier. We constructed the multi-category classifier using SVM classifier. Every category has a classifier like Figure 1. \( classifier_k \) indicates the classifier of category \( c_k \). score\(_k\) is the prediction score of classifier \( classifier_k \).
3.2 Performances of Baselines

We first observe three baselines: TFIDF, OkapiBM25 and the proposed category-based LM.

**TFIDF and OkapiBM25:** They are the most famous word weighting methods in the IR literature. \(tf(w_i,D)\) is the frequency of \(w_i\) in \(D\) and \(df(w_i)\) is the number of document where the \(w_i\) appears as follows:

\[
TFIDF(w_i) = tf(w_i,D) \times idf(w_i)
\]

\[
idf(w_i) = \log \frac{|D|}{df(w_i)} \quad (7)
\]

\[
Okapi(w_i) = \frac{tf_i:(k_1+1)}{tf_i + (1-b) + b \frac{df_i}{avdl}} \cdot \log \left( \frac{n-df_i+0.5}{df_i+0.5} \right) \quad (8)
\]

In equation (8), \(avdl\) is the average document length in the text collection, and \(k_1\) and \(b\) are free parameters, usually chosen, in absence of an advanced optimization, as \(k_1 \in [1.2,2.0]\) and \(b=0.75\). \(n\) is the total number of the documents in the collection. \(tf_i\) is term frequency in the document, and \(df_i\) is the number of document containing \(w_i\).

**Category-based LM:** This type of baseline uses weights of words occurred in each category by equation (1). There are three different types of the probabilities estimated for category classification in the eQA service.

- \(P_{m}(w_i|QA)\) is the occurrence probability of the word on each question-answer pair (QP). This is estimated on equation (1) by replacing \(c_k\) to QA.
- \(P_{m}(w_i|C)\) is the occurrence probability of the word on each category (TP) by equation (1).
- \(P_{m}(w_i|Cor)\) is the occurrence probability of the word on corpus (CP) by equation (5).

These equations represent the role of word in question-answer pair, current category and corpus, respectively. Our model combines them into the following four different LM forms.

- \(LM_QTC\) is the distribution of words using QP, TP and CP.
- \(LM_QT\) is the distribution of words using QP and TP.
- \(LM_TC\) is the distribution of words using TP and CP.
- \(LM_T\) is the distribution of words using only TP.

The performances of these LM combinations are as follows:

**Table 1. Comparison of performances among baselines on REC (%)**

<table>
<thead>
<tr>
<th></th>
<th>Avg.</th>
<th>TFIDF</th>
<th>Okapi</th>
<th>LM_QTC</th>
<th>LM_QT</th>
<th>LM_TC</th>
<th>LM_T</th>
</tr>
</thead>
<tbody>
<tr>
<td>micro</td>
<td>81.9</td>
<td>80.3</td>
<td>81.4</td>
<td>81.4</td>
<td>77.5</td>
<td>81.8</td>
<td></td>
</tr>
<tr>
<td>macro</td>
<td>81.6</td>
<td>79.9</td>
<td>81.0</td>
<td>81.0</td>
<td>76.8</td>
<td>81.2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Avg.</th>
<th>TFIDF</th>
<th>Okapi</th>
<th>LM_QTC</th>
<th>LM_QT</th>
<th>LM_TC</th>
<th>LM_T</th>
</tr>
</thead>
<tbody>
<tr>
<td>micro</td>
<td>81.0</td>
<td>78.5</td>
<td>80.6</td>
<td>80.6</td>
<td>77.3</td>
<td>82.1</td>
<td></td>
</tr>
<tr>
<td>macro</td>
<td>80.2</td>
<td>77.0</td>
<td>79.8</td>
<td>79.7</td>
<td>75.9</td>
<td>81.2</td>
<td></td>
</tr>
</tbody>
</table>

where micro denotes a micro-averaging F1 score, macro a macro-average F1 score. As can be seen from Table 1, TFIDF method outperformed the other baselines on REC; otherwise, the LM_T method outperformed the other baselines on PEC in Table 2.

**3.3 Experimental results of the ratio-based method**

We here evaluate a new word weighting method using the ratio of word occurrence in the current category and the other categories for category classification. Table 3 and 4 show their results.

**Table 3: Comparison of the ratio-based weighting methods on REC (%)**

<table>
<thead>
<tr>
<th></th>
<th>Avg.</th>
<th>TFIDF</th>
<th>LM_T</th>
<th>RateAvg</th>
<th>RateAvgS</th>
</tr>
</thead>
<tbody>
<tr>
<td>micro</td>
<td>81.9</td>
<td>81.8(1.0)</td>
<td>93.7(+11.8)</td>
<td>94.2(+12.3)</td>
<td></td>
</tr>
<tr>
<td>macro</td>
<td>81.6</td>
<td>81.2(-0.4)</td>
<td>93.5(+11.9)</td>
<td>94.1(+12.5)</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4: Comparison of the ratio-based weighting methods on PEC (%)**

<table>
<thead>
<tr>
<th></th>
<th>Avg.</th>
<th>TFIDF</th>
<th>LM_T</th>
<th>RateAvg</th>
<th>RateAvgS</th>
</tr>
</thead>
<tbody>
<tr>
<td>micro</td>
<td>81.0</td>
<td>82.1(+1.1)</td>
<td>90.7(+9.1)</td>
<td>91.1(+10.1)</td>
<td></td>
</tr>
<tr>
<td>macro</td>
<td>80.2</td>
<td>81.2(+1.0)</td>
<td>89.6(+9.4)</td>
<td>90.7(+10.5)</td>
<td></td>
</tr>
</tbody>
</table>

Our experimental results on the RateAvgS method, which employs the additive smoothing, achieved the best F1 score of 94.2% on REC and 91.1% on PEC. The performances of the proposed system have been improved by 12.3% and 10.1% over the TFIDF baseline on REC and PEC, respectively, as shown in
Table 3 and 4. The following Figure 2 and 3 show the performance comparisons among the proposed methods and baselines according to in the number of features; Table 1, 2, 3 and 4 report the experimental results by using the number of features with the best performance.

In this experiment, the number of features was changed from top 10% to 100%, and the total number of feature is 26,420; the chi-square statistics method was used as feature selection. As can be seen from Figure 2 and 3, the changes of curves trend of two proposed methods, RateAvg and RateAvgS, are roughly similar as the number of features increases. And RateAvgS obtained higher performance than RateAvg in all the intervals.

Figure 2. Performance changes according to the number of features on REC (%)  

Figure 3. Performance changes according to the number of features on PEC (%)  

Almost all methods achieved their best performances when using the top 20%-40% of the words ranked by feature selection. The performances of all the methods are convergent after these intervals.

4. Conclusions and Future Work

This paper has studied a new word weighting method for category (or topic) classification in a cQA service by considering the role of word in each category. Our experimental results on the RateAvgS method, which employs the additive smoothing, achieved the best F1 score of 94.2% on REC and 91.1% on PEC. They are 12.3% and 10.1% higher performance in comparison to TFIDF as a baseline system. In addition, we considered two types of category assignment, such as REC and PEC. When PEC is used, precision is increased but recall is reduced. On the other hand, REC has lower precision but higher recall than PEC. The precision of PEC is about 99%. Note that PEC could be more useful method for the cQA service because a user can generally select one from top n ranked categories in the cQA service. These results demonstrated that our proposed method could be effectively used in practical applications.

In future work, we plan to apply the structural information of question-answer pairs to category classification for improving the performance of the cQA service.

5. ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0003780).

6. REFERENCES