2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE)

A Bot for Suggesting Questions that Match Each User's Expertise

Katsunori Fukui
Graduate School of Systems Engineering
Wakayama University
Wakayama, JAPAN
fukui.katsunori@g.wakayama-u.jp

Tomoki Miyazaki
Graduate School of Systems Engineering
Wakayama University
Wakayama, JAPAN
tomoki.miyazaki@g.wakayama-u.jp

Masao Ohira
Graduate School of Systems Engineering
Wakayama University
Wakayama, JAPAN
masao@wakayama-u.ac.jp

***Abstract*—In this extended abstract, we present our approach to the expert recommendation based on PMF (Probabilistic Matrix Factorization) and term expansion for CQA services.**

***Index Terms*—PMF (Probabilistic Matrix Factorization), term expansion, word embeddings, expert recommendation**

I. Introduction

Stack Overflow has been already recognized as an indispensable CQA service for developers, but an important issue has been arising from its popularity. In Stack Overflow, over several thousands of questions and answers are posted on a daily basis. However, about half of the questions are not resolved and about 30% of the questions are not even answered [3]. Furthermore, unresolved questions tend to be increasing these days [1]. Several existing studies [5]–[7] proposed methods to recommend experts (i.e., appropriate questionees for each question) to address the issue above. Our study is also in line with the existing approach, but we are trying to improve the recommendation accuracy using "term expansion" and implementing a recommendation engine as a personal bot in order to help users in CQA services find questions more effectively.

II. Our Approach

Figure 1 shows an overview of our approach which consists of three parts: (A) prediction of each user's voting scores using PMF (Probabilistic Matrix Factorization) [4], (B) keyword expansion using word embeddings, and (C) ranking for the expert recommendation.

A. Prediction of Each User's Voting Scores using PMF

As same as [7], a user-tag expertise matrix is firstly created based on each user's voting scores and tagged keywords used in questions. An entry s_{ij} in the matrix represents an average voting score for user u's answers to questions with a tagged keyword t. Next, PMF is applied to the matrix to predict and complement voting scores in the matrix as illustrated in the top left of Figure 1.

B. Keyword Expansion using Word Embeddings

The existing approach [7] does not consider spelling variants of tagged keywords and similar keywords but deals with them individually. In order to address the issue that makes a user-tag expertise matrix sparse (i.e., the limitation of the tag-based recommendation approach), our approach uses term expansion based on word embeddings. In this study, we use the skip-gram models in Word2vec and fastText respectively. In our approach, keywords tagged for a past question are extracted and are regarded as a set of tagged keywords. Furthermore, a set of tagged keywords is treated as a sentence and all the sets of tagged keywords extracted from past questions in the target dataset are modeled based on word embeddings.

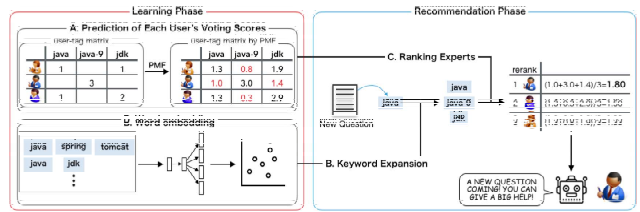

Fig. 1. Overview of our approach

PMF also works as with the term expansion since it complements voting scores for questions which have not been answered by experts. Our approach uses the term expansion to strengthen the effect of PMF using tagged keywords extracted from a newly posted question. The number of expanded terms (keywords) can be arbitrarily changed to optimize the prediction accuracy.

C. Ranking Experts for a Newly Posted Question

For a newly posted question q, the recommendation score $ReScore(u, q)$ is calculated as

$$ReScore(u, q) = \frac{1}{N_t} \sum_{t=0}^{N_t} R(u, t) \quad (1)$$

where $R(u, t)$ represents the recommendation score of user u for tagged keyword t which is calculated by PMF. N_t is the number of keywords tagged for the new question q. $ReScore(u, q)$ means the average score of PMF when user u answers a question q with some tagged keywords. Based on the calculated recommendation scores for all the users, experts are ranked as illustrated in the right of Figure 1.

978-1-7281-2262-5/19/$31.00 ©2019 IEEE
DOI 10.1109/BotSE.2019.00012

III. EXPERIMENT

A. Dataset

For the experiment, questions and answers posted from 2008 to 2017 are collected from Stack Overflow in a XML format as a dataset. Extracting tagged keywords used for questions from July 31, 2008 to December 31, 2014, two word embedding models are created using Word2vec and fastText respectively. For PMF's learning data, tagged keywords from January 1, 2015 to December 31, 2015 are extracted. Voting scores are also extracted only from users who answered over 50 times during the same period. For evaluating our approach and the existing approach, we select questions which are answered by more than six users and their answers from January 1, 2016 to December 31, 2017.

B. Experiment Setting

In the experiment, the existing approach (PMF) [7] and our approach (PMF + term expansion) are compared using the recommendation accuracy. The recommendation accuracy is calculated using nDCG（normalized discounted cumulative gain）[2] which is regularly used as a performance indicator in ranking recommendation studies and indicates how recommended ranked data is adequate. nDCG are expressed from 0 to 1 and the higher nDCG value means the better performance. As our approach has the term expansion feature, we evaluate our approach changing the number of expanded keywords from one to four. For each approach, nDCG is calculated ten times and the average score of them are presented as a result because PMF stochastically predicts and complements values of entities (voting scores) in the user-tag expertise matrix.

C. Result

Table I shows the result of the experiment. The result of the existing approach is used as baseline. From the table, we can confirm our approach using Word2vec outperforms the existing approach in case where additional one (Expand 1) and two (Expand 2) keywords are added to the original keywords tagged to a question. However, the term expansion does not improve the nDCG score in case of using additional three (Expand 3) and four (Expand 4) keywords. On the other hand, our approach using fastText outperforms the existing approach in all the conditions. In particular, one additional keyword by fastText (Expand 1) shows the best nDCG score.

TABLE I
RESULT

Method	nDCG
Baseline	0.824
Word2vec (Expand 1)	**0.825**
Word2vec (Expand 2)	**0.827**
Word2vec (Expand 3)	0.822
Word2vec (Expand 4)	0.823
fastText (Expand 1)	**0.833**
fastText (Expand 2)	**0.825**
fastText (Expand 3)	**0.827**
fastText (Expand 4)	**0.831**

IV. DISCUSSIONS

From the experiment, we found that our approach using the term expansion slightly outperforms the existing approach and the term expansion based on fastText works better than Word2vec. The reason why the term expansion based on fastText is better than Word2vec is that fastText uses information of subwords which are a piece of a word. In general, it is well known that Word2vec cannot precisely obtain synonyms for one word when the word is not frequently appeared in text documents. On the other hand, since fastText learn text data using subwords, the term expansion based on fastText would contribute to obtain similar keywords even from rarely appeared keywords. fastText can also obtain similar keywords which are only included in test data but not included in learning data because of the same reason above (e.g., since keyword "sqlite" not appeared in learning data is divided to sql+qli+lit+ite, similar keywords such as "mysql" and "sql" can be obtained from test data.) , but Word2vec cannot do so.

Although in this paper the term expansion based on word embeddings is assumed to be applied after receiving a new question, we are planing an alternative approach which does not use the term expansion but creates a user-tag expertise matrix based on PMF after unifying similar tagged keywords using word embeddings. Unifying similar tagged keywords in advance might contribute to the improvement in the prediction accuracy and the reduction of the computational cost for PMF since it transforms rarely used keywords (i.e., noisy data for the prediction) into frequently used similar keywords.

V. CONCLUSION AND FUTURE WORK

Our approach based on PMF and the term expansion with Word2vec and fastText slightly outperforms the existing approach only based on PMF. We are currently working to incorporate our approach into an engine for a personal bot which helps expert developer find questions that match their own expertises.

ACKNOWLEDGMENT

This research is conducted as part of Grant-in-Aid for Japan Society for the Promotion of Scientific Research (17H00731, 18K11243).

REFERENCES

[1] M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, and K. A. Schneider, "Answering questions about unanswered questions of Stack Overflow," in *MSR '13*, 2013, pp. 97–100.

[2] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *TOIS*, vol. 20, no. 4, pp. 422–446, 2002.

[3] L. Nie, X. Wei, D. Zhang, X. Wang, Z. Gao, and Y. Yang, "Data-driven answer selection in community QA systems," *TKDE*, vol. 29, no. 6, pp. 1186–1198, 2017.

[4] S. Ruslan and M. Andriy, "Probabilistic Matrix Factorization Ruslan," in *NIPS '07*, 2007, pp. 1257–1264.

[5] L. Wang, B. Wu, J. Yang, and S. Peng, "Personalized recommendation for new questions in community question answering," in *ASONAM '16*, 2016, pp. 901–908.

[6] L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, and Z. Chen, "CQArank: Jointly Model Topics and Expertise in Community Question Answering," in *CIKM '13*, 2013, pp. 99–108.

[7] S. M. Yang Baoguo, "Tag-based expert recommendation in community question answering," in *ASONAM '14*, 2014, pp. 960–963.