Poster: Semantically Enhanced Tag Recommendation for Software CQAs via Deep Learning

Jian Zhang, Hailong Sun, Yanfei Tian, Xudong Liu
SKLSDE Lab, School of Computer Science and Engineering, Beihang University, Beijing, China 100191
Beijing Advanced Innovation Center for Big Data and Brain Computing, Beijing, China 100191
{zjian,sunhl,tianyf,liuxd}@buaa.edu.cn

ABSTRACT

Most software CQAs (e.g. Stack Overflow) mainly rely on users to assign tags for posted questions. This leads to many redundant, inconsistent and inaccurate tags that are detrimental to the communities. Therefore tag quality becomes a critical challenge to deal with. In this work, we propose STR, a deep learning based approach that automatically recommends tags through learning the semantics of both tags and questions in such software CQAs. First, word embedding is employed to convert text information to high-dimension vectors for better representing questions and tags. Second, a Multi-tasking-like Convolutional Neural Network, the core modules of STR, is designed to capture short and long semantics. Third, the learned semantic vectors are fed into a gradient descent based algorithm for classification. Finally, we evaluate STR on three datasets collected from popular software CQAs, and experimental results show that STR outperforms state-of-the-art approaches in terms of $Precision@k$, $Recall@k$ and $F1-Measure@k$.

KEYWORDS

Tag recommendation, deep learning, semantic representation, convolutional neural network, software CQAs, Stack Overflow

1 PROBLEM DESCRIPTION

Software community question-answering sites (CQAs) are becoming increasingly essential for developers to learn and share development knowledge.As a result, a huge amount of users and question-answer entries have been accumulated on those websites. For instance, Stack Overflow, probably the best-known website for developers, has millions of registered users and tens of millions of questions and answers. To efficiently manage such a large amount of information, those Q&A sites usually employ the tagging mechanism to annotate questions. Tags are usually keywords or key phrases with no more than three words in Stack Overflow, which are also metadata of questions. Users can attach at least one but less than or equal to five tags for a question.

However, due to the diversity of developers' technical backgrounds and the freedom of adding tags, it is very common that users cannot always provide correct or consistent tags for the posted questions. Low quality tags are detrimental to software CQA communities. For instance, it can increase the maintenance costs and may affect the quality of search results. This work aims at automating the tagging process to help users choose suitable tags and make tags better managed. Given a question set $Q = \{q_1, q_2, \ldots, q_m\}$ and the corresponding tag set $T = \{t_1, t_2, \ldots, t_n\}$, each question $q_i \in Q$ is associated with a subset of tags $\tau_i = \{t_{k_j} \in T | k \in [1, n], 1 \leq j \leq 5\}$. We refer to the combination of q_i and τ_i as a *question tuple* $q\tau_i = (q_i, \tau_i)$. For each question that have been posted, the corresponding $q\tau_i$ is already known. When a new question q_x is posted, the task is to predict the top-K tags for τ_x.

2 THE STR FRAMEWORK

Some recent efforts [1–3] have studied similar problems, but we propose a novel approach with deep learning. As shown in Fig. 1, the framework of STR mainly includes two phases: training and deployment. In the training phase, question tuples should be prepared first.We preprocess these questions and tags to get word sequences and labels, which is similar to one-hot encoding. During preprocessing, a word dictionary is constructed. The processed question tuples are then fed into the core module, i.e. an artificial neural network, for training. And *Word Embedding* is used to transform each word of a sequence into high dimension vectors. Next, we utilize a multi-task like CNN to extract the different size of semantic features contained in given vectors. Putting it simply, we use both small and large sizes of kernels in convolutional layers to extract the features, which is conducted in parallel before the output layer. The final output layer is a concatenation of the deep semantics learned by different components. We use a gradient descent algorithm to further train the weights by splitting the outputs of CNN into two semantic vectors and taking them as inputs. When making prediction for a new question, we first filter out some unimportant characters and form a word sequence based on the word dictionary. Finally, the trained model is loaded to recommend the top-k tags.

3 EXPERIMENTAL RESULTS

To evaluate the performance of our proposed approach, we conducted a set of experiments on three Stack Overflow datasets: *SO@small*, *SO@medium* and *SO@large* by comparing with *EnTagRec* [1] and *TagMulRec* [3]. The results are shown in Table 1 and 2. When we ran *EnTagRec* on *SO@large*, the program could not finish after over three months' training, so we did not show it in Table 2.

The experimental results demonstrate that STR is superior to state-of-the-art approaches in terms of recommendation accuracy. This can be explained that, deep learning technique usually performs better with increased data and it stores the trained weights to make predictions. The accuracy of *EnTagRec* is more competitive

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05…$15.00
https://doi.org/10.1145/3183440.3194977

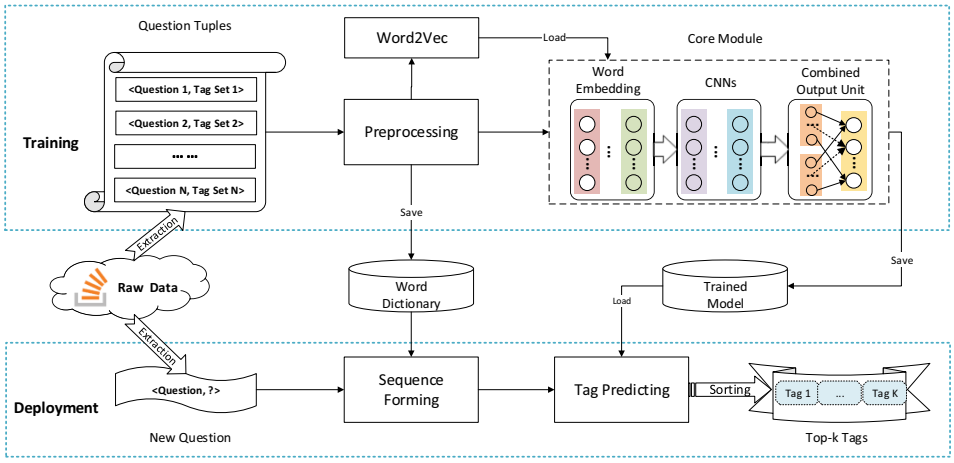

Figure 1: Overall framework of STR

Table 1: Comparison of EnTagRec, TagMulRec and STR on SO@small and SO@medium

	EnTagRec		TagMulRec		STR	
Dataset	SO@small	SO@medium	SO@small	SO@medium	SO@small	SO@medium
Recall@5	80.5	79.3	68.0	73.4	**81.4**	**87.1**
Recall@10	86.8	85.5	77.7	83.6	**89.7**	**94.1**
Average	83.7	82.4	72.9	78.5	**85.6**	**90.6**
Precision@5	**34.6**	35.7	28.4	32.2	34.1	**38.4**
Precision@10	18.7	19.5	16.5	18.6	**19.1**	**21.2**
Average	**26.7**	27.6	22.5	25.4	26.6	**29.8**
F1-Measure@5	46.0	47.2	38.1	43.1	**46.1**	**51.1**
F1-Measure@10	29.1	30.8	26.2	29.6	**30.7**	**33.6**
Average	37.6	39	32.2	36.4	**38.4**	**42.4**
RT(*ms*)	82.41	108.87	7.77	22.06	**1.50**	**1.32**

Table 2: Comparison of TagMulRec and STR on SO@large

	TagMulRec	STR
Recall@5	76.5	**87.8**
Recall@10	85.2	**94.5**
Average	80.9	**91.2**
Precision@5	34.4	**40.0**
Precision@10	19.5	**22.0**
Average	27.0	**31.0**
F1-Measure@5	45.3	**52.4**
F1-Measure@10	30.8	**34.5**
Average	38.1	**43.5**
RT(*ms*)	1,469.18	**1.59**

than *TagMulRec* because it mainly utilizes a Bayes-based component and a complex combination method to predict tags. However, the computational complexity of EnTagRec increases quickly as the number of training samples increases. *TagMulRec* is highly scalable to handle large datasets but lacks accuracy.

4 CONCLUSION

Recommending tags for users is of great importance for software CQAs and we incorporate deep learning to provide a novel solution to this issue. Our experimental results with real-world datasets demonstrate that STR is superior to state-of-the-art methods.

ACKNOWLEDGEMENT

This work was supported partly by National Key Research and Development Program of China (No.2016YFB1000804), partly by National Natural Science Foundation (No 61702024 and 61421003).

REFERENCES

[1] Shaowei Wang, David Lo, Bogdan Vasilescu, and Alexander Serebrenik. 2014. EnTagRec: An Enhanced Tag Recommendation System for Software Information Sites. In *Proceedings of the 2014 IEEE International Conference on Software Maintenance and Evolution (ICSME '14)*. IEEE Computer Society, 291–300.

[2] Xin Xia, David Lo, Xinyu Wang, and Bo Zhou. 2013. Tag Recommendation in Software Information Sites. In *Proceedings of the 10th Working Conference on Mining Software Repositories (MSR '13)*. IEEE Press, 287–296.

[3] Pingyi Zhou, Jin Liu, Zijiang Yang, and Guangyou Zhou. 2017. Scalable tag recommendation for software information sites. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)* . 272–282.