Skill Translation Models in Expert Finding

Arash Dargahi Nobari
Faculty of Computer Science and Engineering Shahid Beheshti University:G.C
Tehran, Iran
a.dargahinobari@mail.sbu.ac.ir

Sajad Sotudeh Gharebagh
Faculty of Computer Science and Engineering Shahid Beheshti University:G.C
Tehran, Iran
s.sotudeh@mail.sbu.ac.ir

Mahmood Neshati
Faculty of Computer Science and Engineering Shahid Beheshti University:G.C
Tehran, Iran
m_neshati@sbu.ac.ir

ABSTRACT

Finding talented users on Stackoverflow can be a challenging task due to term mismatch between queries and content published on it. In this paper, we propose two translation models to augment a given query with relevant words. The first model is based on a statistical approach and the second one is a word embedding model. Interestingly, the translations provided by these methods are not the same. Although the first model in most cases selects pieces of program codes as translations, the second model provides more semantically related words. Our experiments on a large dataset indicate the efficiency of proposed models.

CCS CONCEPTS

• **Information systems** → *Retrieval models and ranking*;

KEYWORDS

Expertise retrieval; Statistical Machine Translation; Semantic matching; Stackoverflow; Talent acquisition

1 INTRODUCTION

Expertise finding is a well-studied field in information retrieval. Several methods have been proposed to solve this problem in bibliographic networks [6], organizations [1] and social networks [7]. In recent years, the increasing availability of big data enables accumulation of evidence of talent and expertise from a wide range of domains. One of such domains is Community Question Answering (CQA) websites such as Stackoverflow which provide users a useful platform for information sharing[5]. In Stackoverflow, users can post questions and answers, leave comments, and provide feedback on the quality of others' posts by voting, commenting and selecting the accepted answer to their questions.

In order to improve the browsing and searching of the questions in Stackoverflow, each question has one or more tags which indicate the required skills to answer that question. These tags can basically be considered as skill areas which recruiters are interested in. For example, consider the following question on Stackoverflow, "What is the difference between JPA and Hibernate?". This question is tagged by the "hibernate", "jpa", "java-ee" and "orm" (i.e. object-relational mapping) tags which are important skill areas in Java language.

The state-of-the-art expert finding language models proposed by Balog *et al.* [1] can be used to rank expert candidates on Stackoverflow. In these models, the associated tags of each question can be considered as queries and the body of answers provided by each candidate is considered as his/her evidence of expertise. The main problem of these models is the vocabulary gap between the textual representation of skills (i.e. tags) and the body of answers provided by candidates. In other words, the exact matching approach proposed in these models fails to bridge this gap. Candidates who are knowledgeable on "orm" usually do not use this word directly in their answers.

Several models in literature have been proposed to overcome the vocabulary gap problem[3]. Specifically, statistical translation models [2], topic modeling [4] and more recently word embedding approaches [9] are among successful approaches to solve this problem.

In this paper, we propose two models to translate a given skill area (e.g. "java-ee", "jpa" and etc.) to a set of relevant words. These translations can help to improve the matching between expert finding queries and the technical textual evidence (i.e. answers) associated with each candidate. These translations can also be used independently by recruiters to detect important aspects of each skill area. For example, the "java-ee" skill area can be translated to *application, web, spring, bean, service, http, session, request, controller and ejb* which are important aspects of "java-ee" in Stackoverflow.

Our first skill translator model (i.e. MI translator) is a statistical model based on mutual information and the second one (i.e. WE translator) is a domain-aware word embedding method which utilizes the specific structure and data of a CQA to translate a skill area to relevant words.

Our experiments on a large dataset generated from Stackoverflow indicate that both the MI and WE methods can improve the MAP measure over language model approaches proposed in [1] and the topic modeling approach proposed in [4].

Interestingly, the translations provided by proposed methods are not the same for a given skill area. Our main finding here is that the MI method provides more specific words (e.g. programming language codes) while WE method selects more human-friendly concepts to translate a given skill area.

SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan

DOI: http://dx.doi.org/10.1145/3077136.3080719

2 PROBLEM SPECIFICATION

As one of the most successful community question answering websites, Stackoverflow is a valuable resource for both software engineers and recruiters. Each question is associated with zero, one or more answers. The first satisfactory answer of a question can be accepted by the questioner and will be highlighted by a green mark on question's web page. The number of accepted answers and the ratio of accepted ones to the whole number of answers provided by a user can be used as a measure of his/her expertise in skill areas (i.e. tags) related to that question. Here, our goal is to find and rank users who are knowledgeable in a given skill area (e.g. "java-ee") when the title and body of questions and answers in a CQA are given.[1]

3 APPROACH

In order to overcome the vocabulary gap between the skill areas and the body of answers, our approach is to translate a given skill area to some relevant words. Then, each answer provided by a candidate containing one of these relevant words is considered as a positive score for that user. Finally, the users are sorted according to sum of their scores for a given skill area. In the following subsections, we explain two methods of skill area translations which are MI (i.e. Mutual Information approach) and WE (i.e. Word Embedding approach), respectively.

3.1 Mutual Information-Based Approach (MI)

Considering each skill area as a class label, the set of answers in CQA can be partitioned into two disjoint subsets. The first subset includes answers tagged by the given skill area, and the second subset includes other answers. In our problem, we can use the Mutual Information (MI) to measure how much information the presence or absence of a term contributes to making the correct classification decision. For each pair of word w and skill area sa, the MI can be calculated using the following equation.

$$MI(sa, w) = \sum_{A_{sa}=0,1} \sum_{A_w=0,1} p(A_{sa}, A_w) \log \frac{p(A_{sa}, A_w)}{p(A_{sa})p(A_w)} \quad (1)$$

In which A_{sa} and A_w are binary variables indicating the event of occurrence of skill area sa and word w in an answer. The probabilities indicated in Equation 1 can be estimated using the following equations:

$$p(A_{sa} = 1) = \frac{c(A_{sa} = 1)}{N}$$
$$p(A_{sa} = 0) = 1 - p(A_{sa} = 1)$$
$$p(A_w = 1) = \frac{c(A_w = 1)}{N}$$
$$p(A_w = 0) = 1 - p(A_w = 1)$$
$$p(A_{sa} = 1, A_w = 1) = \frac{c(A_{sa} = 1, A_w = 1)}{N}$$
$$p(A_{sa} = 1, A_w = 0) = \frac{c(A_{sa} = 1) - c(A_{sa} = 1, A_w = 1)}{N}$$
$$p(A_{sa} = 0, A_w = 1) = \frac{c(A_w = 1) - c(A_{sa} = 1, A_w = 1)}{N}$$
$$p(A_{sa} = 0, A_w = 0) = 1 - p(A_{sa} = 1, A_w = 1) - p(A_{sa} = 1, A_w = 0) - p(A_{sa} = 0, A_w = 1)$$

where $c(A_{sa} = 1)$ indicates number of the answers associated with skill area sa, and $c(A_w = 1)$ indicates number of answers containing word w, and finally N is the number of all answers. To obtain translation probability, the MI score should be normalized using Equation 2. For a given skill area sa, the most informative words can be sorted using $p_{MI}(w|sa)$ probability.

$$p_{MI}(w|sa) = \frac{MI(sa, w)}{\sum_{w'} MI(sa, w')} \quad (2)$$

$p_{MI}(w|sa)$ gives us the probability of translating skill area sa to word w. Intuitively, the probability would be higher if the word w and skill area sa tend to co-occur with each other.

3.2 Word Embedding Based Approach

Topic modeling is one of the most popular techniques that has been successfully applied to solve the vocabulary gap problem. Momtazi *et al.* [4] proposed a method which uses topics extracted from documents as a bridge between queries (i.e. skill areas) and experts to match them. In this method, expert candidates, documents and their terms are mapped to a *topic space* and the matching between them is formulated in the corresponding space.

By reducing the vocabulary gap, the topic modeling approach can improve the retrieval performance in comparison with the document based models proposed in [1]. However, for two reasons, it is necessary to embed document terms and skill areas (i.e. query terms) into a single new space which we call it *skill area space*. First, terms representing skill areas (e.g. "hibernate", "orm" and etc.) rarely occurred in documents and second, a single skill area may be related to more than one topic extracted by topic modeling and conversely a topic may also be related to more than one skill area.

By embedding skill areas and document terms into the same space, in this section, we proposed a domain-aware translation method which maps a given skill area to the most relevant words occurred in documents (i.e. answers of Stackoverflow in our problem). We start by applying topic modeling algorithm to the given set of documents to obtain a low-dimensional representation (i.e. topic space representation) of each word in the dataset. In the next step, we design a mapping function from the topics space to the skill area space. Using this function, the words in documents and the tags representing skill areas can be embedded into a single low-dimensional space as follows: the relevance probability of a skill area sa given a word w can be represented by:

$$p_{WE}(sa|w) = \frac{1}{z} e^{w_{LDA}.W_C + b} \quad (3)$$

In which w_{LDA} is a $1 \times T$ vector representing word w in topic space (T equals to number of the topics), W_C is a $T \times S$ matrix that maps the topic space representation of word w to skill area space (S equals to number of the skill areas), b is a $1 \times S$ vector representing the prior relevance probability of skill areas to a given word and z is the normalization factor.

[1] We use the accepted flag of each answer related to skill areas determined by tags, to define the golden measure. Therefore, we did not use this attribute during test of baselines and proposed algorithms.

In this model, the matrix W_C and vector b are unknown parameters and should be learned during training. We estimate them using error back propagation. During training, for a set of given documents with known tags i.e. skill areas, we estimate the ideal occurrence probability of each word in a given skill area as follows:

$$p_{ideal}(sa|w) = \frac{tf(sa, w)}{tf(w)} \tag{4}$$

where $tf(sa, w)$ is the term frequency of w in documents tagged by sa and $tf(w)$ is the term frequency of w in whole collection. During model constructing, we then optimize the cross entropy of $H(p_{WE}, p_{ideal})$ using batch gradient descent as shown in Equation 5.

$$L(W_C, b) = \frac{1}{m}\sum_{i=1}^{m} H(p_{WE}, p_{ideal}) + \frac{\lambda}{2m}\left(\sum_{i,j} W_{C_{i,j}}^2\right) \tag{5}$$

where $L(W_C, b)$ is the loss function, m is the size of a training batch, and λ is a weight regularization parameter.

Suppose that word w is related to topic t_i and t_j and frequently occurred in the answers related to skill area sa_k and sa_m. During training, the weights of matrix W_C and vector b will be updated such that the representation of word w in skill area space to be placed near to the representation of sa_k and sa_m. In addition, by applying the update rule, other words which are related to t_i and t_j will also get closer to s_k and s_m in skill area space.

As mentioned before, the matrix W_C provides a mapping function from topic space to skill area space. Figure 1 indicates the heat-map of a subset of the matrix after training. Darker cells indicate stronger association between the corresponding topic and skill area and vise versa. For example, skill area SA_4 is more associated with topic T_1 and T_8. While, T_8 is more associated with skill area SA_2 and SA_4. This figure shows the many-to-many relationship between topics and skill areas.

After training the matrix W_C and the vector b, we can estimate the probability of $p(sa|w)$ for each pair of word w and skill area sa using Equation 3. In order to find the most relevant translations for a given skill area sa, we use Bayes' theorem to estimate $p(w|sa) \approx p(w)p(sa|w)$, where $p(w)$ indicates the prior probability of word w to be selected as a good translation. We estimate $p(w)$ using TF-IDF computed over the whole collection in our experiments.

4 EXPERIMENTAL SETUP

4.1 Dataset

Our dataset comes from Stackoverflow[2], covers the period August 2008 until March 2015 containing 24,120,523 posts. In order to reduce the size of dataset, we have selected questions and their associated answers tagged by "java", consisting 2,320,883 total posts, which includes 810,071 questions and 1,510,812 answers.

We mark users as experts on a tag (i.e. skill area) when two conditions are met. First, similar to the definition proposed in [8], they should have ten or more of their answers marked as accepted by the questioner. Second, following the idea proposed in [10], the acceptance ratio of their answers should be higher than the average acceptance ratio (i.e. 40%) in test collection. The first condition filters users with low level of engagement and the second condition filters low-quality users. We select 100 top most frequent tags which co-occurred with "java"' tag in our dataset as expert finding queries. Our queries and implementations for all approaches including baselines is uploaded in github[3].

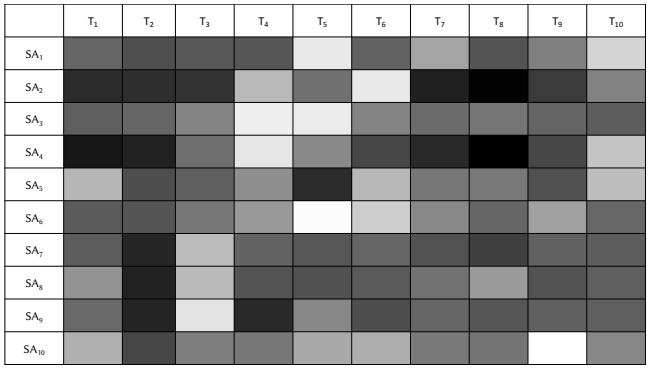

Figure 1: Heat-map of a subset of trained matrix W_C

4.2 Baseline Models

We implemented both Model1 i.e. profile based method referred as LM1 and Model2 i.e. document based method [1] referred as LM2 in the rest of this paper. In our experiments, we used JM smoothing with parameter 0.5 in both methods. In addition, to cope with the vocabulary gap problem, we implemented the topic modeling method (referred as TM in rest of paper) proposed in [4]. The number of topics in our experiments is 100 which equals to the number of skill areas in our dataset. For translation models, we use 20% of available documents as training data.

4.3 Parameter Setting and Implementation Detail

We translate each skill area to top 10 most relevant words using the MI and WE methods described in Section 3.1 and 3.2. Translations are sorted according to their relevance probability for a given skill area. In WE method, we restrict the size of vocabulary to top 2^{16} most-frequent words. In order to optimize the loss function, we use adadelta ($\rho = 0.95$, $\epsilon = 10^{-6}$) with batch gradient descent and weight decay $\lambda = 0.01$. We use Tensorflow [4] to calculate matrix operations on a Nvidia Titan X GPU.

5 RESULTS AND CONCLUSION

Table 2 indicates Mean Average Precision (MAP), P@1 (i.e. precision at first rank), P@5 and P@10 for all methods. According to this table, the TM approach can improve the MAP measure over the LM1 and LM2. By reducing the vocabulary gap, the TM model is able to improve both precision, recall, and accordingly the MAP measure.

According to Table 2, our both translation models perform better than LM1 and LM2 as well as the TM model. Translation models can improve both precision and recall measures. In particular, MI

[2]https://archive.org/details/stackexchange

[3]https://github.com/arashdn/sof-expert-finding/

[4]https://www.tensorflow.org

Table 1: Sample skill area translations using word embedding and mutual information methods

Skill area	Method	Translation 1	Translation 2	Translation 3	Translation 4	Translation 5	Translation 6
hibernate	**MI**	hibernate	entity	table	column	sessionfactory	id
	WE	hibernate	entity	employee	table	query	jpa
swing	**MI**	textsample	jframe	jpanel	jbutton	swing	frame
	WE	jpanel	jbutton	jlabel	jframe	label	frame
selenium	**MI**	__method.apply	selenium	webdriver	driver.findelement	webelement	driver
	WE	tests	junit	test	mock	assertequals	unit
arrays	**MI**	array	int	0	arrays	1	j
	WE	array	index	int	system.out.println	arr	length

Table 2: Performance of baselines and proposed models. * means statistically significant improvement over baseline

Method	MAP	P@1	P@5	P@10
LM 1	37.7	56.0	50.0	44.0
LM 2	36.2	54.0	48.2	42.5
TM	43.4	55.0	53.0	48.8
MI	47.8	**66.0**	60.4	52.9
Improvement vs LM 1	26.8%*	17.9%*	20.8%*	20.2%*
Improvement vs TM	10.1%*	20.0%*	14.0%*	8.4%
WE	**49.6**	65.0	**62.6**	**54.0**
Improvement vs LM 1	31.6%*	16.1%*	25.2%*	22.7%*
Improvement vs TM	14.3%*	18.2%*	18.1%*	10.7%*

model can improve the MAP measure up to 26% over the highest MAP of the language models and up to 10% over the TM model. The WE model has better performance compared to MI model and can improve the MAP measure up to 4% in comparison with MI model.

Table 1 indicates the top six translations for a few number of skill areas extracted by MI and WE models. Interestingly, the MI model usually translates the given skill area to more specific words while WE model selects more general words for the same topic. It seems that the MI model, which is basically a statistical translation model, is more sensitive to the co-occurrence of words and skill areas in documents. As a result, the MI model in most cases selects pieces of program codes (e.g. "__method.apply" for "selenium") which are most frequent words in Stackoverflow answers. On the other hand, the WE model, as a semantic-aware translation model, provides more meaningful and human-friendly translations which can be used in ad-hoc tasks other than expert finding. For example, recruiters can use these translations to select good questions about a skill area.

Figure 2 compares the MAP measure of two translation models with baselines for a different number of translations. According to this figure, by increasing the number of translations, the coverage of a skill area can be increased and accordingly this can improve the MAP measure.

6 FUTURE WORK

The translation models described in this paper provide different relevant words for a given skill area. The next step is to select the best translations among recommended words in order to diversify and maximize the covered sub-topics of a skill area.

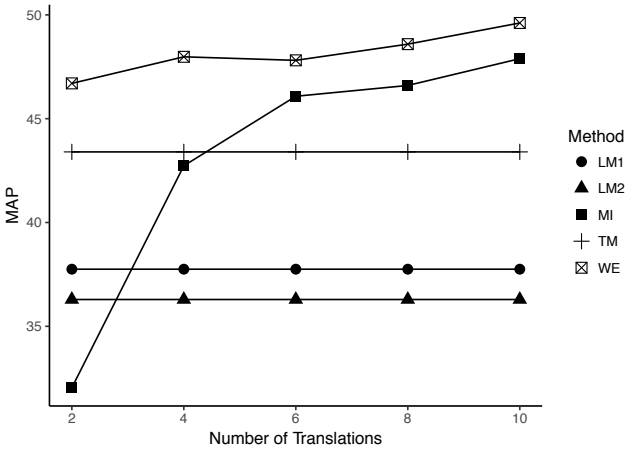

Figure 2: The effect of varying number of translations on MAP measure.

REFERENCES

[1] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. 2009. A language modeling framework for expert finding. *Information Processing & Management* 45, 1 (2009), 1–19.

[2] Maryam Karimzadehgan and ChengXiang Zhai. 2010. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 323–330.

[3] Hang Li, Jun Xu, and others. 2014. Semantic matching in search. *Foundations and Trends® in Information Retrieval* 7, 5 (2014), 343–469.

[4] Saeedeh Momtazi and Felix Naumann. 2013. Topic modeling for expert finding using latent Dirichlet allocation. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery* 3, 5 (2013), 346–353.

[5] Mahmood Neshati. 2017. On early detection of high voted Q&A on Stack Overflow. *Inf. Process. Manage.* 53, 4 (2017), 780–798.

[6] Mahmood Neshati, Seyyed Hadi Hashemi, and Hamid Beigy. 2014. Expertise Finding in Bibliographic Network: Topic Dominance Learning Approach. *IEEE Transactions on Cybernetics* 44, 12 (2014), 2646–2657.

[7] Mahmood Neshati, Djoerd Hiemstra, Ehsaneddin Asgari, and Hamid Beigy. 2014. Integration of scientific and social networks. *World Wide Web* 17, 5 (2014), 1051–1079.

[8] David van Dijk, Manos Tsagkias, and Maarten de Rijke. 2015. Early Detection of Topical Expertise in Community Question Answering. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*. 995–998.

[9] Christophe Van Gysel, Maarten de Rijke, and Marcel Worring. 2016. Unsupervised, efficient and semantic expertise retrieval. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1069–1079.

[10] Jie Yang, Ke Tao, Alessandro Bozzon, and Geert-Jan Houben. 2014. Sparrows and Owls: Characterisation of Expert Behaviour in StackOverflow. In *User Modeling, Adaptation, and Personalization - 22nd International Conference, UMAP 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings*. 266–277.