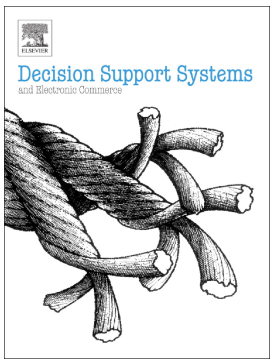
Journal Pre-proof



An improvement in the quality of expert finding in community question answering networks

Mahdi Dehghan, Ahmad Ali Abin, Mahmood Neshati

PII: S0167-9236(20)30180-9

DOI: https://doi.org/10.1016/j.dss.2020.113425

Reference: DECSUP 113425

To appear in: *Decision Support Systems*

Received date: 5 May 2020

Revised date: 1 September 2020

Accepted date: 13 October 2020

Please cite this article as: M. Dehghan, A.A. Abin and M. Neshati, An improvement in the quality of expert finding in community question answering networks, *Decision Support Systems* (2020), https://doi.org/10.1016/j.dss.2020.113425

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Published by Elsevier.

Journal Pre-proof

An Improvement in the Quality of Expert Finding in Community Question Answering Networks

Mahdi Dehghan[a], Ahmad Ali Abin[a,*], Mahmood Neshati[a]

[a]*Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran*

Abstract

Expert finding in Community Question Answering (CQA) networks such as Stack Overflow is a practical issue facing a challenging problem called vocabulary gap. A widely used approach to overcome this problem is translation model. Different from prior works that only consider the relevancy of translations to a query, we intend to diversify query translations for better coverage of query topics. In this work, we have utilized the idea of clustering to group relevant translations to a given query into different clusters and then select representatives from each cluster as a set of diverse translations. We have proposed two new approaches to cluster translations. In the first one, the Mutual Information was primarily utilized as a similarity measure during clustering. In the second approach, the relevant translations are embedded in a topic space and then clustered in that space. After clustering, we propose two batch and sequential methods to select a diverse set of translations from the resultant clusters. The batch method selects the top most relevant translations from each cluster proportional to the relevancy of that cluster to the user query. The sequential one is an iterative method that looks for the most diverse set of translations considering the previously selected ones. Finally, to rank users, a regression model was utilized to learn how expert and non-expert users differ in using a set of diverse translations in their documents. Experiments on a large dataset generated from Stack Overflow demonstrate that the proposed methods improve the ranking performance over baselines in the expert finding.

Keywords: Expert finding, Question answering, Stack Overflow, Translations diversification

2010 MSC: 00-01 99-00

* Corresponding author

Email addresses: mahdi.dehghan551@gmail.com (Mahdi Dehghan), a_abin@sbu.ac.ir (Ahmad Ali Abin), a_neshati@sbu.ac.ir (Mahmood Neshati)

Journal Pre-proof

1. Introduction

Expert finding is an well-studied problem in Information Retrieval (IR) research which has attracted the attention of many researchers [1, 2]. Given a user query, the focus of expert finding is to address the task of providing an ordered list of experts in the field of the given query. Finding talented people has found many applications in Community Question Answering (CQA) networks [3, 4, 5]. Stack Overflow is one the most popular CQA websites. In Stack Overflow, users can post questions and answers, leave comments and determine the importance and quality of posts by voting and choosing the accepted answer. Any questioner should assign one or more tags to his question to identify the skills needed for answering the question.

The tags associated with each question are considered as queries and a set of answers given by users are used as textual expertise evidences. Finding talented people in Stack Overflow is a challenging task due to the term mismatch between the query and textual expertise evidence (i.e. answers in Stack Overflow) of candidates [4]. In fact, an expert in a specific tag like "OSX" domain, rarely uses the word "OSX" in his answers. Instead, he uses a set of words (such as "mac", "os", "apple", etc.) that make the concept of "OSX" more specialized.

So far, multiple research studies has been conducted to overcome the vocabulary gap issue by using translation models [6, 4, 7]. Although translation models have been successful in solving the VG problem, they are not very accurate in many applications. There are two main reasons for this: 1) they did not cover query topics as much as possible. Our experiments indicate that the more a set of translations cover important query topics, the less the effect of the VG will be, 2) they did not efficiently utilize a set of translations to estimate the expertise of candidates.

In this work, we investigate the effectiveness of translations diversification on the quality of expert finding. To this end, we extract relevant translations to each query in the first step. Next, we cluster the translations to extract a set of representative as a diverse set of translations. Finally, for a better ranking of candidates, we construct a regression model to learn how expert and non-expert users differ in using a set of diverse translations in their documents. Experiments on a large dataset gathered from Stack Overflow show the effectiveness of the proposed methods against the baselines.

1.1. Related work

Journal Pre-proof

Expert finding as a well-studied field in information retrieval, addresses the task of identification and ranking knowledgeable people in the subject of user query. Several models in literature have been proposed to solve this task [8], [9]. Expert finding was studied in multiple domains including organizations [6, 10], bibliographic networks [11, 12, 13], CQA networks [3, 1, 14], social networks [15, 16] and etc.

Neshati *et al.* [17] have addressed the task of future expert finding with respect to the expertise evidence in the current time in Stack Overflow. They proposed a supervised learning framework in order to predict the best ranking of experts in the future. They have defined four different feature groups; namely, topic similarity, emerging topics, user behavior, and topic transition and examined the impact of them in the future expert finding problem.

Sotudeh *et al.* [14] have mined the shape of expertise and defined three knowledge levels including Advanced, Intermediate, and Beginner which are obtained by the ranking of users in a specific context. Further, they have defined 3 types of users: Non-expert, T-shaped, and C-shaped. With regard to the knowledge levels and the list of expertise that a user possesses, then they have determined the type of each user in the dataset. Finally, by proposing three approaches (DBA, EBA, and XEBA), they have evaluated the effectiveness of each approach on IR metrics.

Dehghan *et al.* [18] have proposed a new method for T-shaped expert finding on Stack Overflow. They have taken the temporal dynamics of expertise into account to mine the shape of expertise of each candidate expert by using an LSTM neural network. They have applied a filtering technique on top of the LSTM neural network to create a profile for each candidate exert. Finally, They have determined the shape of expertise for each candidate by using his profile.

One of the challenging problems in the CQA websites is the problem of VP which was previously investigated by Dargahi Nobari *et al.* in [4] and Dehghan *et al.* in [3]. Topic modeling [19], translation models [3, 4, 20] and word embedding [21] are considered as efficient solutions to overcome the VG problem.

Topic modeling is one of the well-known techniques that has been utilized in a large number of text mining tasks [22, 23]. Tang *et al.* [24] have proposed a novel topic modeling algorithm, named as Labeled Phrase Latent Dirichlet Allocation (LPLDA) considers simultaneously semantical label information and the ordering of words. Momtazi *et al.* [19] have proposed an approach based on topic modeling to find experts in TREC Enterprise track for 2005 and 2006. They have extracted the main topics of documents in the collection using the Latent Dirichlet

Allocation method [25]. Then, they have used these topics in a probabilistic framework as an intermediary to rank candidates.

Karimzadehgan *et al.* [20] have proposed a statistical translation model to overcome the problem of VG between queries and documents. They have considered the normalized mutual information between two words as translation probability. They have also proposed a regularization of self-translation probabilities in order to overcome the problem of under-estimating self-translation (i.e. translate a word into itself) probabilities.

Dargahi *et al.* [4] have also proposed two translation models to overcome VG problem. They have translated query words into a set of relevant words during the translation stage and then used these translations to predict the best ranking of experts at the stage of score aggregation. Their first translation model was based on normalized Mutual Information and their second one was based on word embedding. In the score aggregation phase, they have used query translation to retrieve relevant documents and then ranked the candidates according to the total number of retrieved relevant documents. In these translation models, VG problem was somewhat resolved. However, the issue of diversification in translations has been ignored. Moreover, in the score aggregation phase, they have used a naive model to rank candidates. Authors in [2] utilized other features to improve the results. Accordingly, they have introduced the concept of vote share to determine the quality of the posts. Eventually, the posts with higher vote share would have more impact on the translation model.

Dehghan *et al.* [3] have proposed a translation model to overcome the VG problem. In the first step, they have clustered important words in a query space. Then, each cluster in the query space is clustered again in a co-occurrence space. Representatives of each cluster in the co-occurrence space are returned to their parent cluster in the query space. Finally, candidate experts are retrieved and ranked using query translations which are selected using a probabilistic model.

To sum up, the previous works on expert finding problems can be categorized into two main groups. The first one [6, 26, 27]-which is older in comparison with the second group- mainly focused on the modeling of the expert finding and completely ignored VG (i.e. query term mismatch) problem. The second group proposed naive methods to overcome the vocabulary mismatch problem. Specifically, Momtazi and Naumann [19], proposed a topic modeling approach and Dargahi et.al [4] and Dehghan et.al [3] proposed a translation approach to solve VG problem.

Basically, in these methods, the user query is expanded with some few terms to reduce the gap between query and document terms. Although, these methods have been successful to improve the quality of expert finding but they ignore the diversification of selected terms to cover all aspects of the user query. Our paper is a natural extension of the second group to improve the quality of expert finding by selecting relevant as well as diverse translations for the user query.

1.2. Contributions of this work

The main contributions of this work are as follows:

1. We investigate the effectiveness of translation diversification approach on the quality of expert finding. In other words, we attempt to diversify query translations besides considering the relevancy of translations to a user query in order to overcome VG problem.
2. We use two clustering approaches and a regression model to learn how to effectively diversify the query translation.
3. We experiment on a large dataset gathered from Stack Overflow and show that the proposed methods can significantly outperform the state of the art [6, 19] as well as the best performing translation models [4, 3, 2] in terms of Mean Average Precision (MAP).

1.3. Organization of this paper

In Section 2, we will present the proposed method for expert finding. Details for the dataset, baseline models, evaluation measures, parameter setting, and implementation are given in Section 3. Discussion was presented in Section 4. The paper concludes with conclusions and future directions in Section 5.

2. The proposed method

Existing works in translation models only considered the relevancy of translations to the user's query during expert finding. Such approaches could not provide acceptable results when the coverage of query topics by the relevant translations is low. In order to involve diversification in translation models, we retrieve relevant translations at first step, then we utilize the idea of data

clustering to group the retrieved translations into different clusters and select representatives from the clusters as a set of diverse translations. The general idea behind the proposed method was summarized in Algorithm 1. As the algorithm shows, the proposed method ranks candidates in four steps. In the first step, it extracts a set of relevant translations T_q^{rel} for a given query q from D, where D is a set of documents (see section 2.1). In the second step, T_q^{rel} is grouped into k_q different clusters $C_q = \{c_q^1, \ldots, c_q^{k_q}\}$. Here we have two different approaches to do this, clustering based on MI (see section 2.2.1) and clustering in topic space (see section 2.2.2). In the next step, a diverse set of translations T_q^{div} is selected from C_q. Here, we have two different approaches to do this, batch (see section 2.3.1) and sequential (see section 2.3.2). Finally, the proposed method ranks candidates using T_q^{div} (see section 2.4). Due to the earlier explanations, we have four different translation models that consider both relevancy and diversity in translating of a given query. Further details of the proposed methods will be explained in following sections. Also, Table 1 indicates the summary of the notations of this work.

Algorithm 1 The proposed expert finding method. **Input**: Query q. **Output**: Sorted list of experts E_q.

1: **procedure** RANKEXPERTS(q)
2: $D \leftarrow$ Set of documents (answers) in CQA.
3: **Step 1**: $T_q^{\text{rel}} \leftarrow$ Extract relevant translations of q from D.
4: **Step 2**: $C_q \leftarrow$ Cluster T_q^{rel}.
5: **Step 3**: $T_q^{\text{div}} \leftarrow$ Extract diverse translations from C_q.
6: **Step 4**: $E_q \leftarrow$ Rank candidates using T_q^{div}.
7: **return** Sorted list of experts E_q.

Table 1: Summary of notations.

Notation	Notation description
q	query
$w_\bullet$	word

Journal Pre-proof

$\boldsymbol{x}$	word $w_\bullet$ or query q
$\boldsymbol{T_q^{rel}}$	Set of relevant translations of query q
$\boldsymbol{T_q^{div}}$	Set of diverse translations of query q
$\boldsymbol{D}$	Set of documents (Answers) in CQA
$\boldsymbol{C_q}$	Set of resultant clusters due to clustering set of relevant translations T_q^{rel}
$\boldsymbol{c_q^i}$	i^{th} cluster in C_q

2.1. Step 1: Extraction of relevant translations

In the first step, we extract top-n most relevant translations for a query q (i.e. T_q^{rel}). We calculate mutual information (MI), which is a well-known measure for determining how much information the presence or absence of a term contributes to the relevancy of two words [4, 3, 20], between query q and each word $w_\bullet$ in the set of documents D. We then normalize MI score to obtain $p_{MI}(w_\bullet \mid q)$ which is the translation probability of query q into the word $w_\bullet$ as shown in the following equation.

$$p_{MI}(w_\bullet \mid q) = \frac{MI(q, w_\bullet)}{\sum_{w'_\bullet} MI(q, w'_\bullet)} \tag{1}$$

where $MI(q, w_\bullet)$ is calculated as the following.

$$MI(q, w_\bullet) = \sum_{A_q=0,1} \sum_{A_{w_\bullet}=0,1} p(A_q, A_{w_\bullet}) \log \frac{p(A_q, A_{w_\bullet})}{p(A_q)\, p(A_{w_\bullet})} \tag{2}$$

where A_q and $A_{w_\bullet}$ are two binary variables indicating the event of occurrence of q and w in a document (i.e. answer) $d \in D$. At the end of this step, we have extracted a set of translations T_q^{rel} that is highly relevant to query q.

2.2. Step 2: Clustering relevant translations

In the second step, we cluster T_q^{rel} into k_q different clusters $C_q = \{c_q^1, \ldots, c_q^{k_q}\}$. Here, the idea is to group similar translations into same cluster for a better translations diversification. To this

Journal Pre-proof

end, we propose two methods to cluster T_q^{rel}. In the following, each method is discussed in more details.

2.2.1. Method 1: Clustering based on MI

In this solution, for each query q we group T_q^{rel} into different clusters of translations by utilizing MI as similarity measure. We refer to this method as CMI in the rest of the paper. It is obvious that different clusters are representatives for different query topics, and semantically similar translations will be grouped in the same cluster. By choosing translations from different clusters, we expect to cover more query topics. So, the first step in this solution is to determine how much the words in T_q^{rel} are semantically similar. To this end, we utilize $p_{\text{MI}}(w_j \mid w_i)$ which is the translation probability of word w_i into w_j as similarity measure during clustering. The value of $p_{MI}(w_j \mid w_i)$ is estimated by using the following equation.

$$p_{MI}(w_j \mid w_i) = \frac{MI(w_i, w_j)}{\sum_{w' \in T_q^{\text{rel}}} MI(w_i, w')} \tag{3}$$

where $MI(w_i, w_j)$ is calculated in the following way.

$$MI(w_i, w_j) = \sum_{A_{w_i}=0,1} \sum_{A_{w_j}=0,1} p(A_{w_i}, A_{w_j}) \log \frac{p(A_{w_i}, A_{w_j})}{p(A_{w_i})\, p(A_{w_j})} \tag{4}$$

where binary variables A_{w_i} and A_{w_j} indicate the event of occurrence of words w_i and w_j in document (i.e. answer) $d \in D$. Having the pairwise similarity of translations $p_{MI}(w_j \mid w_i)$ for all $w_i, w_j \in T_q^{\text{rel}}$ calculated using Eq. (3), we cluster T_q^{rel} into k_q groups of translations using Algorithm 2. As the algorithm shows, the proposed clustering method selects a random word $w_\bullet$ from among T_q^{rel} at the first step, then it calculates the similarity of $w_\bullet$ to each cluster $c_q^i \in C^q, i = 1, ..., k_q$ and selects the cluster with maximum similarity as the winner cluster in the second step. Finally if the maximum similarity is higher than a predefined threshold then the word $w_\bullet$ will be assigned to the winner cluster, otherwise it will remain as a member of its cluster. This process will be completed when no change exists in the clusters assignment or the total number of

epochs exceeds a maximum epoch. After clustering, we expect that each cluster $c_q^i, i = 1,...,k_q$ contains translations that are semantically similar and cover same query topics.

Algorithm 2 The method proposed for translations clustering based on MI. **Input**: Set of n top most relevant translations T_q^{rel} to query q, The similarity matrix $P \equiv [p_{ij}]_{n \times n}$ where p_{ij} is the probability of translating word $w_i \in T_q^{\text{rel}}$ into $w_j \in T_q^{\text{rel}}$ (i.e. $p_{\text{MI}}(w_j \mid w_i)$) and the input parameter β that controls when a new cluster should be created. **Output**: Set of clusters $C_q = \{c_q^i\}, i = 1,...,k_q$

---

procedure CLUSTERING($P, T_q^{\text{rel}}, \beta$)

$T_q' \leftarrow T_q^{\text{rel}}$

$w_\bullet \leftarrow$ Select a random word from T_q'.

$c_q^1 \leftarrow c_q^1 \cup \{w_\bullet\}$

$T_q' \leftarrow T_q' / \{w_\bullet\}$ ▷ Remove word $w_\bullet$ from T_q'

Initialize the set of clusters C_q with c_q^1.

$loop = 1$

while $loop \neq maxLoop$ **and** cluster assignments change **do**

while T_q' isn't empty

$w_\bullet \leftarrow$ Select a random word from T_q'.

$T_q' \leftarrow T_q' / \{w_\bullet\}$ ▷ Remove word $w_\bullet$ from T_q'

$c_q^{winner} \leftarrow arg\,max_{c \in C_q} \dfrac{\sum_{\substack{u \in c \\ u \neq w_\bullet}} P(u, w_\bullet)}{|c|}$ ▷ $|c|$ means number of words in cluster c

$s \leftarrow \max_{c \in C_q} \dfrac{\sum_{\substack{u \in c \\ u \neq w_\bullet}} P(u, w_\bullet)}{|c|}$

if $s >= \beta$ **then**

Journal Pre-proof

Remove $w_\bullet$ from its previous cluster c_q^{prev}.

16: $c_q^{winner} \leftarrow c_q^{winner} \cup \{w_\bullet\}$

else

18: Create new cluster c_{new}.

20: $c_q^{new} \leftarrow c_q^{new} \cup \{w_\bullet\}$

$C_q \leftarrow C_q \cup \{c_q^{new}\}$

$loop \leftarrow loop + 1$

22: $T_q' \leftarrow T_q^{\text{rel}}$

return Set of clusters C_q.

2.2.2. Method 2: Clustering in topics space

The second method for clustering is an embedding approach that clusters relevant translations in a new space consisting of main topics of query (This method referred to as CTS in the rest of this paper). In this method, in order to group relevant translations into different clusters, we create a new space of main topics of the given query, then embed words in that space. Two words with close vectors in this space are more relevant than the ones with far vectors. We embed relevant translations of the given query by estimating their relevancy to the main topics of the given query. Therefore, the first step is to extract main topics of query q and translations $w_i \in T_q^{\text{rel}} : i = 1, 2, ..., n$ in order to embed words in query main topic space. Given any word x (x may be a query q or a translation w_i), we extract the main topics of x in the following way: In the first step, we use LDA algorithm in order to group documents $d_i \in D : i = 1, 2, .., N$ into M topics to estimate the probability of topic $z_i \in Z : i = 1, ..., M$ given x (i.e. $\hat{p}(z_i \mid x)$). Having $\hat{p}(z_i \mid x)$, we extract the main topics covered by x using Eq. (5). This equation labels topic z_i as main topic for x (i.e. x is highly related to z_i) if the probability of z_i given x is higher than the prior probability of z_i [28].

$$\mathcal{J}(x, z_i) = \begin{cases} 1 & if \ \hat{p}(z_i \mid x) > p(z_i) \\ 0 & otherwise \end{cases} \tag{5}$$

Journal Pre-proof

Let $A \in \mathbb{R}^M$ be a column vector where being 1 at each position j means that topic z_j is a main topic for query q and $A' \in \mathbb{R}^M$ be another column vector where being 1 at each position l means that the topic z_l is a main topic of translation w_i. Let $m = \sum_{j=1}^{M} A[j]$ be the total number of main topics extracted for query q. For example, consider we have grouped all documents into 5 topics $z_1, z_2, ..., z_5$. Assuming z_2 and z_4 are the main topics of a particular query q and z_2, z_3 and z_5 are the main topics of a particular translation w_i, the corresponding topics indication vector $A = [\underset{z_1}{0}, \underset{z_2}{1}, \underset{z_3}{0}, \underset{z_4}{1}, \underset{z_5}{0}]^T$ and $A' = [\underset{z_1}{0}, \underset{z_2}{1}, \underset{z_3}{1}, \underset{z_4}{0}, \underset{z_5}{1}]^T$.

After extracting main topics for query q and translations $w_i \in T_q^{\text{rel}} : i = 1, 2, ..., n$, we embed each translation w_i into a new $(m+1)$ -dimensional vector $V = [v_1, ..., v_m, v_{m+1}]^T$, where $v_o : o = 1, .., m$ identifies the coverage of w_i on the topic corresponding to o^{th} non-zero element of A. The last dimension $v_{m+1} = 1 - \sum_{o=1}^{m} v_o$ indicates the amount of non-coverage of query topics by w_i. Calculation of $[v_1, ..., v_m, v_{m+1}]^T$ is done as indicated in following equations:

$$v_o = \begin{cases} p(z_{idx} \mid w_i) & o \neq m+1, A[idx] = 1, A'[idx] = 1 \\ 0 & o \neq m+1, A[idx] = 1, A'[idx] = 0 \\ \sum_{\substack{l=1 \\ A'[l]=1}}^{M} p(z_l \mid w_i) & o = m+1 \end{cases} \tag{6}$$

where idx is the position of o^{th} non-zero topic in A and $p(z_{idx} \mid w_i)$ is the probability of topic z_{idx} given w_i which is normalized on main topics of w_i and calculated by using Eq. (7):

$$p(z_{idx} \mid w_i) = \frac{\hat{p}(z_{idx} \mid w_i)}{\sum_{\substack{j=1 \\ A'[j]=1}}^{M} \hat{p}(z_j \mid w_i)} \tag{7}$$

After embedding each translation w_i into the new space, we group them in k_q clusters $c_q^i, i = 1, ..., k_q$ which are used for translation diversification in the next step.

Journal Pre-proof

2.3. Step 3: Translations diversification

In the third step, we try to select most diverse set of translations T_q^{div} from $C_q = \{c_q^1, \ldots, c_q^{k_q}\}$ (i.e. the result of clustering obtained from previous step). Here, the idea is to increase the coverage of query topics by choosing a diverse set of translations from C_q. By this, we hope the final accuracy of expert finding to be enhanced by choosing experts with a wider coverage of query topics. To this end, we propose two batch and sequential methods to select a diverse set of translations from C_q.

2.3.1. Method 1: Batch translations diversification

In the batch strategy, to select a diverse set of translations T_q^{div} from the resultant clusters $C_q = \{c_q^1, \ldots, c_q^{k_q}\}$, we choose some representative translations from each cluster. The number of representatives selected from each cluster c_q^i is determined by using the following equation.

$$share^{c_q^i} = \left\lfloor \frac{\sum_{w_\bullet \in c_q^i} p_{MI}(w_\bullet \mid q)}{\sum_{j=1}^{k_q} \sum_{w_\bullet \in c_q^j} p_{MI}(w_\bullet \mid q)} \times n' \right\rceil \tag{8}$$

where [.] operator rounds the input argument to its nearest integer and n' is the total number of translations that we tend to diverse for each query. In fact, sharing translations gives more chance to the more relevant clusters to be presented in the final set of translations. After determining the share of each cluster $c_q^i, i = 1, ..., k_q$, the words of that cluster are sorted in descending order based on their translation probabilities into the query (i.e. $p_{MI}(ww_\bullet \mid q)$) and the $share^{c_q^i}$ top words are returned as the representatives (i.e. diverse translations selected from c_q^i) of that cluster.

2.3.2. Method 2: Sequential translations diversification

The sequential method for translations diversification is based on concepts in topic modeling. In this method, we iteratively choose an optimal translation to cover important query

Journal Pre-proof

topics which are less covered by the previously selected translations. Let $w_1, w_2, ..., w_{t-1} \in T_q^{\text{div}}$ be the previously selected translations for query q and w_t be the t^{th} candidate translation. At each step t, we utilize the main topics extracted for query q and translations $w_i : i = 1,..,t$ to select the optimal translation. The main topics for query q and translations $w_i : i = 1,..,t$ are extracted by utilizing the method described in Section 2.2.2.

Let $A_x \in \mathbb{R}^M$ be a column vector in which being 1 at each position l indicates that the topic z_l is a main topic of word x (where x may be query q or a translation $w_i : i = 1,..,t$) and let $\hat{p}(z_j \mid x)$ be the probability of any topic $z_j : j = 1,...,M$ given x which is calculated during extraction of main topics. We calculate the normalized probability of each main topic z_l given word x (i.e. $p(z_l \mid x)$) using the following equation:

$$p(z_l \mid x) = \frac{\hat{p}(z_l \mid x)}{\sum_{\substack{j=1 \\ A_x[j]=1}} \hat{p}(z_j \mid x)} \tag{9}$$

The proposed method for sequential translations diversification is illustrated in Algorithm 3. As this algorithm shows, we select the most relevant translation from each cluster and score it by using Eq. (10). Then we select translation with the maximum score (referred to as the winner translation) as the first best and remove it from the winner cluster. Next, the best translations are selected sequentially in the following way. At each iteration, we select the most relevant translation from each cluster and choose from among them the translation with the maximum relevancy to one of the query topics (i.e. $p(z \mid q)\, p(w_\bullet \mid q, z)$) and the minimum relevancy to the topics covered by the previously selected translations (i.e. $\prod_{\{w'_\bullet \in T_q^{\text{div}}\}} (1 - p(w'_\bullet \mid q, z))$).

$$S(q, w_t, T_q^{\text{div}}) = \underbrace{\lambda \sum_{\substack{j=1 \\ A_q[j]=1}}^{M} \left(p(z_j \mid q)\, p(w_t \mid q, z_j) \prod_{\{w'_\bullet \in T_q^{\text{div}}\}} (1 - p(w w'_\bullet \mid q, z_j)) \right)}_{\text{Diversification Score}} + (1-\lambda) \underbrace{p_{MI}(w_t \mid q)}_{\text{Relevancy Score}} \tag{10}$$

where λ controls the amount of diversification, $p_{MI}(w_t \mid q)$ indicates the translation

probability of word w_t to query q calculated by Eq. (3), $p(z|q)$ shows the relevancy of query q to topic z calculated by using Eq. (9), and $p(w_\bullet|q,z)$ shows how much the word $w_\bullet$ covers topic z from query q which is obtained using the following equation:

$$p(w_\bullet|q,z) \propto p(w_\bullet|q) \times p(z|w_\bullet) \tag{11}$$

where $p(z|w_\bullet)$ is calculated using Eq. (9), and $p(w_\bullet|q) = \frac{2\times\left(n - R_{w_\bullet} + 1\right)}{n\times\left(n+1\right)}$, where $R_{w_\bullet}$ is the ranking of word $w_\bullet$ in T_q^{rel} and n is the total number of initial translations obtained in Section 2.1.

Algorithm 3 The proposed method for sequential translations diversification. **Input**: Query q, set of clusters $C_q = \{c_q^1, \ldots, c_q^{k_q}\}$ and n', totoal number of translations that we want to diverse for each query. **Output**: Set of diverse translations T_q^{div}

---

procedure SEQUENTIALDIVERSIFICATION (q, C_q, n')
  $T_q^{\text{div}} = \emptyset$
  for $i = 1 \ldots n'$ **do**
    $maxScore \leftarrow 0$
    for $j = 1 \ldots k_q$ **do**
      $w_\bullet \leftarrow$ The most relevant translation selected from cluster c_q^j
      $score \leftarrow Score(q, w_\bullet, T_q^{\text{div}})$ By using Eq. (10)
      if $score > maxScore$ **then**
        $maxScore \leftarrow score$
        $winnerTranslation \leftarrow w_\bullet$
        $idx \leftarrow j$ ▷ Keep the index of winner cluster
    $T_q^{\text{div}} \leftarrow T_q^{\text{div}} \bigcup \{winnerTranslation\}$
    $c_q^{idx} \leftarrow c_q^{idx} / \{winnerTranslation\}$
  return Set of diverse translations T_q^{div}.

Journal Pre-proof

2.4. Step 4 : Ranking of candidates

The ranking is an important step playing a significant role in the accuracy of expert finding. During the ranking, we utilize the query translations to estimate the expertise of candidates. The ranking method proposed in [4], referred to as *Naive method*, is a successful model in the expert ranking. For each query q, a Naive ranking method retrieves all documents containing at least one word from its translations (T_q^{rel}) in the first step. Next, it scores candidates based on the number of retrieved documents that he/she authored. Finally, it sorts candidates based on their scores [4] and returns them as the final ranked results.

The ranking method presented in [4], does not consider the fact that experts and non-expert users may differ in the way they are using expertise keywords in their documents. In this section, a new ranking approach is proposed which tries to discover how expert and non-expert users differ in using a set of discriminative keywords in their documents. Therefore, for each query q, we consider its diverse translations T_q^{div} as the discriminative keywords and a training set is constructed based on the behavior of expert and non-expert users in answering (writing documents) in the following way. For each user having at least one answer, we will create a training vector based on term frequency (TF) of discriminative keywords (diverse translations T_q^{div}) in his answers. If he is expert in query q, his training vector will be labeled as 1 indicating the pattern of keywords generated by an expert. Otherwise, that vector will be labeled as 0 showing the user does not use such a pattern for keywords usage in his answers. After generating the training data, we train a multi-layer neural network $\aleph_q$ for each query q, which consists of an input layer with the number of neurons equals to the total number of diverse translations n', a hidden layer containing ten neurons and an output layer with a single neuron.

We use the trained network $\aleph_q$ during expert ranking in the following way: For each candidate in query q, we create a test vector based on the term frequency of discriminative keywords (diverse translations T_q^{div}) in his answers. The output of the trained network $\aleph_q$ shows the expertise degree of that candidate in the range of 0 to 1. By sorting outputs in descending order, the quality of experts ranking can be evaluated in terms of MAP, P@K, and so on.

Journal Pre-proof

To sum up, as we described in the Section 2, there are four different combinations of the proposed method that consider both relevancy and diversity in translating a given query. We explain when to use which combination of the proposed method will produce more desired results. Both CMI and CTS are utilized to create boundaries between the relevant translations. In CMI we utilize a clustering approach to create these boundaries but in CTS we first embed each word in a topic space and then cluster each translation in the new main topic space of query. Actually, CTS is more intelligent because we consider relevancy to query during making these boundaries. In other words, in CTS we consider how much two translation are similar to each other and to the query word but in the CMI we only consider the relevancy of two translations to each other. Considering relevancy to query can be beneficial for creating relevant translations. After clustering relevant translations, we proposed two different approaches to select some representatives from clusters. In the first one, which is a batch method, we choose the most relevant translations from each cluster. Share of each cluster in selecting representatives are specified based on its relevancy to the query. In the second one, which is a sequential approach, we first select the most relevant translation from each cluster. After that we calculate a score for each one based on its relevancy to the query and how it covers main query topics. Eventually, the translation with maximum score is selected. We keep doing this process until selecting enough amount of translations. As it is obvious the second approach is more intelligent but it is a little time consuming than the batch method. Therefore, we expect to get better result in the presence of CTS method and sequential approach.

3. Experimental setup

In this section, the effectiveness of the proposed method is investigated in comparison with state-of-the-art approaches in the expert finding. Details for the dataset, baseline models, evaluation measures, parameter setting, and implementation are given in the following sections.

3.1. Dataset

We have used a subset of questions and their corresponding answers posted to Stack Overflow[1] between August 2008 and March 2015 in the domain of "JAVA" and "PHP" as the dataset for expert finding. Both "JAVA" and "PHP" are two popular domains that including a large

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

Journal Pre-proof

number of documents. This dataset includes $810,071$ questions and $1,510,812$ answers related to "JAVA" domain and $714,476$ questions and $1,298,107$ answers related to "PHP" domain. The generated dataset consists of 200 tags which have highly occurred with "JAVA" or "PHP" tags. To be more specific, we have considered 100 top tags which have highly occurred with "java" and 100 top tags having most co-occurrence with "PHP" as candidate queries for expert finding. To label experts in a particular tag, two conditions must occur simultaneously [3]: having at least 10 accepted answers and, acceptance ratio higher than average acceptance ratio in dataset (40% in both "JAVA" and "PHP" domains [3]). The test collection will be uploaded on Github[2].

3.2. Baseline Models

We have compared the proposed method against eight well-known methods in expert finding literature. The first baseline method is a profile-based language model (referred to as LM1 in the rest of this paper) and the second one is a document-based language model (referred to as LM2 in the rest of paper) [6]. Both LM1 and LM2 are set up with the JM smoothing parameter equals to 0.5. The third and fourth baseline methods [4] are two models which are selected owing to the fact these approaches are considering VG issue in expert finding which makes them a sufficient baseline to compare the proposed approaches with. These methods are based on word embedding and mutual information and referred to as WE and MI, respectively. The fifth and sixth baselines method [2] are extensions of two proposed translations methods in [4] which are referred to as WE(VS) and MI(VS). The seventh baseline method [3]is a clustering translation model referred to as CTM in the rest of this paper. CTM tries to overcome the VG problem by diversifying query translations to improve the quality of final ranking. The last baseline method [19] is based on topic modeling referred to as TM in the rest of this paper. TM also tries to cope with VG problem in expert finding. However, it uses a different approach.

3.3. Evaluation Measures

In this section, the evaluation metrics for evaluating the quality of results are introduced. We have used Coverage, Confidence and F_{score} metrics [29] to compare the quality of translations.

[2] www.github.com

The more topics a set of translations covers, the better the result of expert ranking will be (i.e. Coverage). When two sets of translations cover the same query topics, the set with a larger number of translations per query topic is preferred (i.e. Confidence).

Let $T_q^{\bullet}$ be a set of translations for a given query q. The coverage measure of $T_q^{\bullet}$ is defined as percentage of query topics (here, the query q) covered by $T_q^{\bullet}$. To calculate the topic coverage of $T_q^{\bullet}$ we need to determine how much a translation $w \in T_q^{\bullet}$ covers the main topics of q. Accordingly, in the first step we group documents $d_i : i = 1,2,..,N$ into M topics $z_j \in Z : j = 1,..,M$ using LDA algorithm and extract main topics of w as illustrated in Eq. (5). After extracting the main topics, we calculate the coverage of $T_q^{\bullet}$ using the following equation.

$$Coverage(T_q^{\bullet}, q) = \frac{\sum_{z_j \in Z} \left(\mathcal{J}(q, z_j) \mathbf{1}_{\{\sum_{w_j \in T_q^{\bullet}} \mathcal{J}(w_j, z_j)\}} \right)}{\sum_{z_j \in Z} \mathcal{J}(q, z_j)} \tag{12}$$

where $\mathbf{1}_{\{condition\}} = 1$ if $condition > 0$. Also, we take the confidence of $T_q^{\bullet}$ into account to measure the redundancy of translations in covering query topics. Confidence measure is defined in the following way:

$$Confidence(T_q^{\bullet}, q) = \frac{\sum_{z_i \in Z} \frac{\mathcal{J}(q, z_i) \sum_{w_j \in T_q^{\bullet}} \mathcal{J}(w_j, z_i)}{n}}{\sum_{z_i \in Z} \left(\mathcal{J}(q, z_i) \mathbf{1}_{\{\sum_{w_j \in T_q^{\bullet}} \mathcal{J}(w_j, z_i)\}} \right)} \tag{13}$$

where $\mathbf{1}_{\{condition\}} = 1$ if $condition > 0$. The last measure is $\mathrm{F}_{\mathrm{score}}$ that is defined as the harmonic mean of coverage and confidence as follows:

$$F_{score} = \frac{2 \times Coverage(T_q^{\bullet}, q) \times Confidence(T_q^{\bullet}, q)}{Coverage(T_q^{\bullet}, q) + Confidence(T_q^{\bullet}, q)} \tag{14}$$

In order to assess the quality of experts ranking, the precision at k ($p@k$) and Mean Average Precision (MAP) that are two commonly used evaluation measures for ranking in IR, are

Journal Pre-proof

employed in this work. The evaluation measure $p@k$ for a given query q is the percentage of experts in the top k retrieved results.

$$p@k = \frac{\text{number of experts in the top } k \text{ retrieved results}}{k} \tag{15}$$

The MAP evaluation measure is defined as the mean value of Average Precision (AP) for all queries in test collection. For each query q, we define AP by using the following relation:

$$AP = \frac{\sum_{k=1}^{|E_q|} p@k \times rel(k)}{R} \tag{16}$$

where E_q is ordered list of candidates for query q, $|E_q|$ indicates the total number of candidates, R is the total number of real experts in the golden set, and $rel(k)$ is a binary function indicating the expertise of given candidate.

3.4. Parameters Setting and Implementation Details

To prepare Stack Overflow data for the next processing step, we did some preprocessing on data to remove HTML tags and scripts, remove stop-words, stem words occurred in documents, extract topics and so on. The Apache Lucene[3] Standard Analyzer was used to stop-words removal and words stemming. The well-known MALLET topic modeling package was used for topic extraction from the body of posts in the collection. The implementation of all retrieval models was done by using the Apache Lucene tool.

The first step in the proposed method is to extract n top most relevant translations T_q^{rel} to query q (Section 2.1). T_q^{rel} will be diversified in the next steps to select a set of diverse translations T_q^{div}. In this paper, we extract top-200 relevant translations as T_q^{rel} which are candidates for diversification (i.e. $n = 200$). Because the proposed method takes both relevancy and diversity of translations into account, any word that is not highly relevant to the query has a lower chance to be presented at the final set of translations. It is a nice feature because the proposed method prevents the presence of translations unrelated to the query on the final set of translations. When we increase n, the irrelevant translations will have more chances to come up with the final

[3] https://lucene.apache.org

Journal Pre-proof

results and this contradicts the basic principle of translation models.

In the first proposed method for clustering (i.e. CMI method described in Section 2.2.1), we have the parameter β which is the minimum value of similarity for assigning data to clusters. When β is greater than a maximum threshold, the number of clusters will be equal to the number of translations, and when its value is lower than a minimum threshold, all translations will be grouped into a single cluster. In such cases, if we use the batch method for translations diversification (Section 2.3.1) the diversity of translations is ignored. As a result, the best value of parameter β should be initialized between a minimum and maximum threshold. To estimate the maximum threshold, we average the maximums of pairwise similarities between translations for all queries. Similarly, to estimate the minimum threshold, we average the minimums of pairwise similarities between translations for all queries. Table 2 shows maximum and minimum thresholds for β in "JAVA" and "PHP" domains. Then, we fine-tune β in range of minimum to the maximum threshold for all 100 tags in "JAVA" and select the best value of β based on average coverage, average confidence and average F_{score} in order of priority from first to last. Similarly, the optimal value for β in "PHP" is calculated. The optimal value of β for tags in "JAVA" and "PHP" domains are 0.01 and 0.04, respectively.

Table 2: Maximum and minimum thresholds for β in "JAVA" and "PHP" domains.

Domain	Minimum threshold	Maximum threshold
PHP	0.0006	0.06
JAVA	0.0003	0.05

In the second proposed method for translation clustering (i.e. CTS method described in Section 2.2.2), the well-known k-Means clustering algorithm is utilized to cluster relevant translations T_q^{rel} in the embedding space. As is known in the literature, finding the optimal number of clusters for the k-Means algorithm has remained as an open issue in machine learning research [30]. A common test to determine the appropriate number of clusters is to plot the ratio of the between-group variance to the total variance against the various number of clusters and determine the place of "elbow" where the slope of the curve is leveling off to the right of the plot. To the right of this point, there is not an impressive reduction in variance when the number of clusters increases.

In this work, we use the elbow test to determine the optimal number of clusters for relevant translations T_q^{rel} in the embedding space and set the optimal number of clusters to the value of k that is corresponding to 90% percentage of variance.

Each neural network used for ranking candidates (Section 2.4) is configured with 10 neurons in the hidden layer with the RELU activation function. The activation function of the output layer is selected to be a sigmoid function in which the output is in a range of 0 to 1. The weights of each network are learned using the stochastic batch gradient descent method with a batch size of 50 and the learning rate of 0.01. We use Tensorflow[4] to create and train these networks.

3.5. Experimental results

To evaluate the effectiveness of proposed method, three experiments are conducted. The effect of diversification on the quality of translations is studied in the first experiment. In the second one, the effectiveness of the proposed method in expert finding is compared with baseline methods introduced in Section 3.2. Because we have proposed two clustering methods for translations (i.e CMI and CTS) and two methods for translation diversification (i.e. batch and sequential), the notation $\mathcal{C}+\mathcal{D}$ is used to show a combination of clustering method $\mathcal{C}$ and diversification method $\mathcal{D}$. In the last experiment, we investigate the effectiveness of the proposed blending method (Section 2.4) on various sets of translations. In what follows, each experiment is described in more details.

3.5.1. Experiment 1: The effect of diversification on the quality of translations

In this experiment, we study the effect of diversification on the quality of translations. First, we show several examples of translations diversified by the proposed method in Figure 1. In this figure, the word cloud visualization technique is used to expeditiously identify the relevancy and diversification of translations to the query word. We asked an expert to realize that each translation is placed in which query topics. In Figure 1 the words on the same topic, are displayed with the same color. As a result, the existence of more colors in each word cloud demonstrates covering more query topics. Moreover, the selection order of translations is visualized by font size such that

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

the earlier translations have a larger font size. To better illustrate the selection order of translations, the rating of each translation is also included. As demonstrated in this figure, the diversification of colors in each word cloud shows that the suggested method has been able to include a high degree of diversification.

Figure 1: Sample query translations using CMI+Sequential method in "JAVA" domain.

To compare the proposed and baseline methods in terms of translation quality, for each baseline and proposed method we report the average Coverage, average Confidence and average F_{score} for 100 tags in "JAVA" and 100 tags in "PHP". The results are summarized in Table 3. As shown in this table, the proposed method outperforms the baseline methods on translation quality. A noteworthy point in Table 3 is that the quality of translations increases when the sequential method is used for translation diversification.

Table 3: Comparison of the proposed method over the baseline methods in terms of ability to cover query topics.

Tag	Method	Average Coverage	Average Confidence	Average F_{score}
PHP	MI [4]	73.2	49.8	48.1
	WE [4]	74.4	53.6	52.8
	CMI+Batch	76.2	54.2	52.6
	CTS+Batch	76.7	60.2	56.2
	CMI+Sequential ($\lambda = 0.8$)	77.0	**60.5**	56.6
	CTS+Sequential ($\lambda = 0.8$)	**78.0**	59.3	**57.7**
JAVA	MI [4]	82.8	50.7	56.5
	WE [4]	83.0	51.9	56.9
	CMI+Batch	85.3	51.9	57.0
	CTS+Batch	86.2	56.4	61.2

	CMI+Sequential ($\lambda = 0.7$)	87.2	57.6	63.1
	CTS+Sequential ($\lambda = 0.7$)	**87.7**	**57.8**	**63.6**
CMI ≡ Translations **C**lustering based on **M**utual **I**nformation				
CTS ≡ Translations **C**lustering in **T**opic **S**pace				
Batch ≡ **Batch** translations diversification				
Sequential ≡ **Sequential** translations diversification				

An important parameter which can affect the quality of translations in sequential method for diversification (Section 2.3.2) is λ. Increasing λ will increase coverage and confidence until it exceeds a threshold by which both coverage and confidence are significantly diminished. That is because in such a situation the proposed sequential method will be more interested in choosing diverse translations without considering their relevancy. The effect of different values of λ on coverage and confidence are plotted in Figure 2 for tags in "PHP" domain.

Figure 2: The effect of different values of λ on coverage and confidence in "PHP" domain.

We also investigate the effect of the number of translations on the quality of experts ranking for all baselines and the best-proposed method (i.e. CTS+Sequential). Fig. 3-a and 3-b indicate the sensitivity of the best-proposed method (i.e. CTS+sequential) and all baselines on the different number of translations for both "Java" and "PHP" domains, respectively. Evidently, almost in all methods, by increasing the number of query translations, the quality of expert ranking will improve. It can be explained by considering that increasing the number of query translations makes more query topics coverage and improve the quality of expert ranking. In the proposed method, we try to select a diverse set of query translations to cover more query topics. So we expect that increasing the number of query translations makes more improvement in the proposed method than the baseline methods.

Figure 3: Analysis the effect of varying number of translations on MAP measure for all baselines and the best proposed method (i.e CTS+Sequential).

3.5.2. Experiment 2: The effectiveness of diversification in the expert finding

In this experiment, the effectiveness of the proposed method in expert finding is compared with the baseline methods. The results are summarized in Table 4. As this table shows our proposed methods have significantly improved ranking metrics over the baseline models. A remarkable point in this table is that in both "JAVA" and "PHP" domains, the proposed method outperforms the baseline methods. Another point is that the sequential method for expert finding is performing better than the batch and all other baseline methods in terms of MAP, $P@1$, $P@5$, and $P@10$ measures.

Table 4: Comparison of the proposed method with the baselines methods.

Tag	Method	MAP	AVG. P@1	AVG. P@5	AVG. P@10
PHP	LM 1 [6]	37.7	56.0	50.0	44.0
	LM 2 [6]	36.2	54.0	48.2	42.5
	TM [19]	40.1	53.0	55.0	49.1
	MI [4]	45.8	59.0	61.2	56.1
	WE [4]	50.3	60.0	62.6	58.1
	CTM [3]	53.2	61.2	62.5	58.7
	MI(VS) [2]	58.7	75.0	72.6	64.2
	WE(VS) [2]	56.2	75.0	69.6	62.1
	CMI+Batch	67.6	72.0	69.4	58.1
	CTS+Batch	67.7	72.0	69.8	59.6
	CMI+Sequential ($\lambda = 0.8$)	68.1	72.0	69.8	61.3
	CTS+Sequential($\lambda = 0.8$)	**70.0**	**76.0**	**71.6**	**62.6**
JAVA	LM 1 [6]	37.7	56.0	50.0	44.0
	LM 2 [6]	36.2	54.0	48.2	42.5
	TM [19]	43.4	55.0	53.0	48.8
	MI [4]	47.8	66.0	60.4	52.9

Journal Pre-proof

	WE [4]	49.6	65.0	62.6	54.0
	CTM [3]	52.2	69.2	61.5	55.8
	MI(VS) [2]	64.7	85.0	73.6	65.2
	WE(VS) [2]	66.0	86.0	72.8	66.1
	CMI+Batch	75.8	82	75.3	64.8
	CTS+Batch	75.9	83	77.9	67.0
	CMI+Sequential ($\lambda = 0.7$)	77.2	83.0	77.7	67.0
	CTS+Sequential($\lambda = 0.7$)	**78.1**	**86.0**	**78.3**	**67.5**
CMI ≡ **C**lustering based on **M**utual **I**nformation					
CTS ≡ **C**lustering in **T**opic **S**pace					
Batch ≡ **Batch** translations diversification					
Sequential ≡ **Sequential** translations diversification					

Table 5 reports improvement of the best-proposed method (i.e. CTS+Sequential) over the baselines. As illustrated in this table, the proposed CTS+Sequential method has a significant improvement over the baseline methods.

Table 5: Improvement of the best proposed method (i.e. CTS+Sequential) over the baselines.

Tag	Methods	MAP	P@1	P@5	P@10
PHP	CTS+Sequential vs TM	74.5%	43.3%	30.1%	27.4%
	CTS+Sequential vs MI	52.8%	28.8%	16.9%	11.5%
	CTS+Sequential vs WE	37.5%	26.6%	14.3%	7.7%
JAVA	CTS+Sequential vs TM	79.9%	56.3%	47.7%	38.3%
	CTS+Sequential	63.3%	30.3%	29.6%	27.5%

Journal Pre-proof

	vs MI				
	CTS+Sequential vs WE	57.4%	32.3%	14.3%	15.9%

3.5.3. Experiment 3: Investigating the effectiveness of the ranking method

In this experiment, we investigate the effectiveness of the proposed method for experts ranking (Section 2.4) over a various set of translations. Due to using a different method for expert ranking, we evaluate the effectiveness of baseline methods by replacing the proposed ranking method with their built-in ranking method to have a fair comparison of results (see Table 6).

Table 6: Performance of baselines utilizing our proposed ranking method.

Tag	**Translation method**	**Ranking method**	**MAP**	**P@1**	**P@5**	**P@10**
PHP	MI [4]	Proposed	59.1	69.0	63.9	53.3
		Built-in[4]	45.8	59.0	61.2	56.1
	WE [4]	Proposed	63.0	75.0	67.3	58.1
		Built-in[4]	50.9	60.0	62.6	58.1
JAVA	MI [4]	Proposed	70.3	76.0	70.6	61.4
		Built-in[4]	47.8	66.0	60.4	52.9
	WE [4]	Proposed	72.0	79.0	73.0	64.6
		Built-in[4]	49.6	65.0	62.6	54.0

In order to show the relationship between Coverage and *MAP*, we have plotted these measures in Figures 4-(a) and 4-(b) for all tags in "PHP" and "JAVA" domain, respectively. The *MAP* of all baseline methods in these figures are calculated using the proposed ranking method instead of the built-in method in order to make a fair comparison. As these figures show, the more topics are covered by a set of translations, the more precise the expert finding method will be. This fact justifies the idea behind the proposed method in the diversification of translations in order to improve the results.

Figure 4: The relationship between topics covered by a set of translations and the MAP measure in:

a) “PHP”, and b) “JAVA” domain.

The paper also makes mention of common transformations such as damping the effects of raw counts using sub-linear functions, therefore it is possible to determine the degree to which suboptimal feature shaping might be contributing to the need for feature selection. By this, it is easy to learn much that is generalizable from the results that are presented. To this end, we demonstrate the effectiveness of the best-proposed method (i.e. CTS+Sequential) when the square root, cube root, and logarithm of feature vectors are used for damping effect analysis in the proposed ranking method. The results are given in Table 7.

Table 7: Performance analysis of the best proposed method (i.e. CTS+Sequential) encountering damping effects of raw counts using sub-linear functions.

Tag	**Sub-linear function**	**MAP**	**P@1**	**P@5**	**P@10**
PHP	$\log_2(\bullet)$	67.9	73.0	70.1	61.3
	$\sqrt[2]{\bullet}$	68.2	73.0	70.9	62.2
	$\sqrt[3]{\bullet}$	67.1	73.0	69.8	61.8
JAVA	$\log_2(\bullet)$	76.7	83.0	78.9	69.1
	$\sqrt[2]{\bullet}$	77.8	83.0	79.7	68.0
	$\sqrt[3]{\bullet}$	76.3	84.0	79.1	68.2

Journal Pre-proof

4. Discussion

As we described in Section 2, there are four different combinations of the proposed method that consider both relevancy and diversity in translating a given query. Both CMI and CTS find boundaries between the relevant translations. A clustering approach was used by CMI to find the boundaries but the clustering of translations embedded in the new topic space was used by CTS for the boundaries detection. Experiments show tah CTS is more smart than CMI because it considers the relevancy to query during boundaries detection. In other words, CTS considers how much translation are similar to each other and to the query word, but CMI only considers the relevancy of translations to each other. After clustering the relevant translations, we use two batch and sequential

method to select representative for each cluster. In the batch method, we choose the most relevant translations from each cluster and the share of each cluster in selecting representatives are specified based on its relevancy to the query. In the sequential method, we first select the most relevant translation from each cluster and calculate a score for each one based on its relevancy to the query and how it covers the main query topics. Experiments show that the second one is more intelligent but it is a little time consuming than the batch one. Therefore, we expect to get better result in the presence of CTS and sequential approach.

In the following, we do an analysis on the computational complexity of the proposed method. Let $T(n,m)$, be the computational complexity of the proposed method, where n and m stands for the total number of vocabularies and documents in the collection, respectively. As mentioned in the section 2, we extract top-k relevant translations, where $k \le n$, by using MI criteria in the first step of the proposed method. The time complexity of this step is bounded to the total number of vocabulary (i.e. $O(nm + n\log(n))$). In the next step of the proposed method, we cluster top-k extracted translations. We have introduced two CMI and CTS clustering approaches for this step. The time complexity of CMI is bounded to number of selected relevant translations and number of documents (i.e. $O(km + k^2)$). In CTS approach we have utilized K-means clustering algorithm, therefore the time complexity of this approach is $O(k^2)$. Eventually, in the last step of our proposed translation model we intend to select a set of diverse translations that cover query topics as much as possible. To do so, we have introduced two Batch and Sequential approaches. The time complexity of Batch and Sequential approaches are $O(k\log(k))$ and $O(k\log(k)+k)$, respectively. Therefore, if we consider $k = n$, the time complexity of the best proposed approach (i.e. CTS + Sequential) will be $O(nm + n\log(n) + n^2 + n\log(n) + n) \sim O(nm + n^2)$. As it is obvious from Table 4, the best baseline methods in *PHP* and *JAVA* domains are MI(VS) and WE(MS), respectively. The time complexity of MI(VS) is $O(nm + n\log(n))$ which is relatively close to the best combination of the proposed method. The best baseline method in *JAVA* domain is a MLP neural network based approach which as the results shows is not reliable in all domains and has a big problem that is if a user poses a new query which is not in the predefined set of translations, it can not translate that. Therefore, although the proposed method is slightly more time-consuming, it performs much better in terms of the quality of results and has relieved the shortcomings of baseline methods to some extent.

Journal Pre-proof

5. Conclusion

Expert finding in CQA websites has attracted the attention of many researchers. VG is a basic problem in CQA websites making expert finding a challenging task. In this work, translation models are used in a way that takes both relevancy and diversity into account during translating queries. The mutual information measure has been used to ensure that translations are relevant to the user's query and the idea of clustering has been used to consider diversification of query translations. For a better ranking of candidates, a regression model was used to learn how expert and non-expert users differ in using a set of diverse translations in their documents. Our experiments indicate that diversifying query translations can be beneficial in the quality of translations and consequently ranking of candidates.

References

[1] M. Dehghan, H. A. Rahmani, A. A. Abin, V.-V. Vu, Mining shape of expertise: A novel approach based on convolutional neural network, Information Processing & Management 57 (4) (2020) 102239.

[2] A. D. Nobari, M. Neshati, S. S. Gharebagh, Quality-aware skill translation models for expert finding on stackoverflow, Information Systems 87 (2020) 101413.

[3] M. Dehghan, A. A. Abin, Translations diversification for expert finding: A novel clustering-based approach, ACM Trans. Knowl. Discov. Data 13 (3) (2019) 32:1–32:20.

[4] A. Dargahi Nobari, S. Sotudeh Gharebagh, M. Neshati, Skill translation models in expert finding, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2017, pp. 1057–1060.

[5] P. Rostami, M. Neshati, T-shaped grouping: Expert finding models to agile software teams retrieval, Expert Systems with Applications 118 (2019) 231–245.

[6] K. Balog, L. Azzopardi, M. de Rijke, A language modeling framework for expert finding, Information Processing & Management 45 (1) (2009) 1–19.

[7] H. Li, J. Xu, et al., Semantic matching in search, Foundations and Trends® in Information Retrieval 7 (5) (2014) 343–469.

[8] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, L. Si, et al., Expertise retrieval, Foundations

Journal Pre-proof

and Trends® in Information Retrieval 6 (2–3) (2012) 127–256.

[9] S. Lin, W. Hong, D. Wang, T. Li, A survey on expert finding techniques, Journal of Intelligent Information Systems 49 (2) (2017) 255–279.

[10] Q. Wang, J. Ma, X. Liao, W. Du, A context-aware researcher recommendation system for university-industry collaboration on r&d projects, Decision Support Systems 103 (2017) 46–57.

[11] S. H. Hashemi, M. Neshati, H. Beigy, Expertise retrieval in bibliographic network: a topic dominance learning approach, in: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, ACM, 2013, pp. 1117–1126.

[12] C. Moreira, P. Calado, B. Martins, Learning to rank academic experts in the dblp dataset, Expert Systems 32 (4) (2015) 477–493.

[13] C. Moreira, A. Wichert, Finding academic experts on a multisensor approach using shannons entropy, Expert Systems with Applications 40 (14) (2013) 5740–5754.

[14] S. S. Gharebagh, P. Rostami, M. Neshati, T-shaped mining: A novel approach to talent finding for agile software teams, in: European Conference on Information Retrieval, Springer, 2018, pp. 411–423.

[15] M. Neshati, D. Hiemstra, E. Asgari, H. Beigy, Integration of scientific and social networks, World wide web 17 (5) (2014) 1051–1079.

[16] Y. Xu, D. Zhou, J. Ma, Scholar-friend recommendation in online academic communities: An approach based on heterogeneous network, Decision Support Systems 119 (2019) 1–13.

[17] M. Neshati, Z. Fallahnejad, H. Beigy, On dynamicity of expert finding in community question answering, Information Processing & Management 53 (5) (2017) 1026–1042.

[18] M. Dehghan, M. Biabani, A. A. Abin, Temporal expert profiling: With an application to t-shaped expert finding, Information Processing & Management 56 (3) (2019) 1067–1079.

[19] S. Momtazi, F. Naumann, Topic modeling for expert finding using latent dirichlet allocation, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 3 (5) (2013) 346–353.

[20] M. Karimzadehgan, C. Zhai, 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,

2010, pp. 323–330.

[21] C. Van Gysel, M. de Rijke, M. Worring, Unsupervised, efficient and semantic expertise retrieval, in: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 1069–1079.

[22] X. Li, C. Li, J. Chi, J. Ouyang, Short text topic modeling by exploring original documents, Knowledge and Information Systems 56 (2) (2018) 443–462.

[23] C. Li, W. K. Cheung, Y. Ye, X. Zhang, D. Chu, X. Li, The author-topic-community model for author interest profiling and community discovery, Knowledge and Information Systems 44 (2) (2015) 359–383.

[24] Y.-K. Tang, X.-L. Mao, H. Huang, Labeled phrase latent dirichlet allocation and its online learning algorithm, Data Mining and Knowledge Discovery 32 (4) (2018) 885–912.

[25] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of machine Learning research 3 (Jan) (2003) 993–1022.

[26] M. Neshati, E. Asgari, D. Hiemstra, H. Beigy, A joint classification method to integrate scientific and social networks, in: European Conference on Information Retrieval, Springer, 2013, pp. 122–133.

[27] M. Neshati, H. Beigy, D. Hiemstra, Expert group formation using facility location analysis, Information processing & management 50 (2) (2014) 361–383.

[28] F. M. Belém, C. S. Batista, R. L. Santos, J. M. Almeida, M. A. Gonçalves, Beyond relevance: explicitly promoting novelty and diversity in tag recommendation, ACM Transactions on Intelligent Systems and Technology (TIST) 7 (3) (2016) 26.

[29] M. Karimzadehgan, C. Zhai, G. Belford, Multi-aspect expertise matching for review assignment, in: Proceedings of the 17th ACM conference on Information and knowledge management, ACM, 2008, pp. 1113–1122.

[30] A. A. Abin, A random walk approach to query informative constraints for clustering, IEEE Transactions on Cybernetics (2018) 1–12.

Journal Pre-proof

Credit Author Statement:
Mahdi Dehghan: Data Curation, Software, Writing - Original Draft
Ahmad Ali Abin: Conceptualization, Methodology, Supervision, Writing - Review & Editing
Mahmood Neshati: Conceptualization, Validation, Review

Journal Pre-proof

- Translations diversification was investigated on the quality of expert finding
- Both diversity and relevancy of translations were considered for expert finding
- A regression model was used for expert ranking

Journal Pre-proof

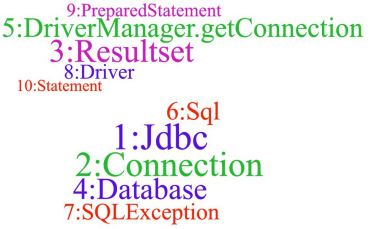
Biographical Note

Mahdi Dehghan received the B.Sc. in computer engineering from the University of Golestan, Golestan, Iran, in 2016. He received the M.Sc. degrees in computer engineering from the Shahid Beheshti University, Tehran, Iran, in 2019. His primary research interest is in the area of information retrieval and machine learning.

Ahmad Ali Abin received the B.Sc. in computer engineering from the Iran University of Science and Technology, Tehran, Iran, in 2005. He received the M.Sc. and Ph.D. degrees in computer engineering from the Sharif University of Technology, Tehran, Iran, in 2008 and 2014, respectively. Since 2014, he joined with the Department of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran. His research interests include pattern recognition, machine learning and neural computing.

Mahmood Neshati received the B.S. (2005), M.S. (2007) and PhD (2014) degrees in computer engineering from the Sharif University of Technology, Tehran, Iran. He is currently an assistant professor at Shahid Beheshti University, Tehran, Iran. Prior to that, he was a research assistant at Qatar Computing Research Institute (QCRI-2015). He has published several research papers in Information Retrieval journals and conferences (SIGIR, CIKM and ECIR). His main research interests also include big data analytic, large scale distributed system design and information management.

Journal Pre-proof



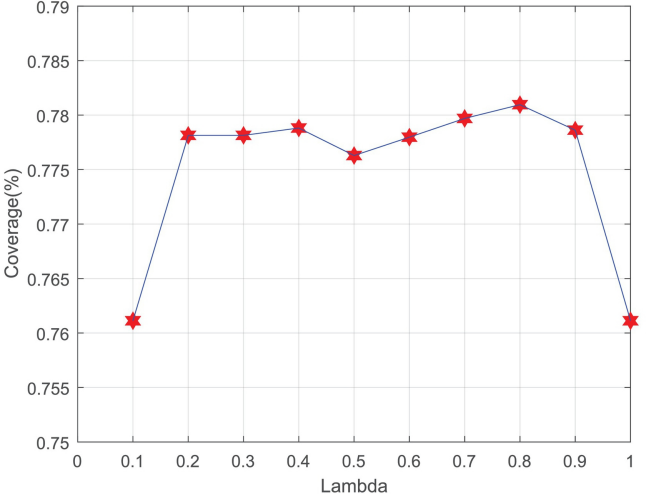
(a) Jdbc

(b) Hibenate

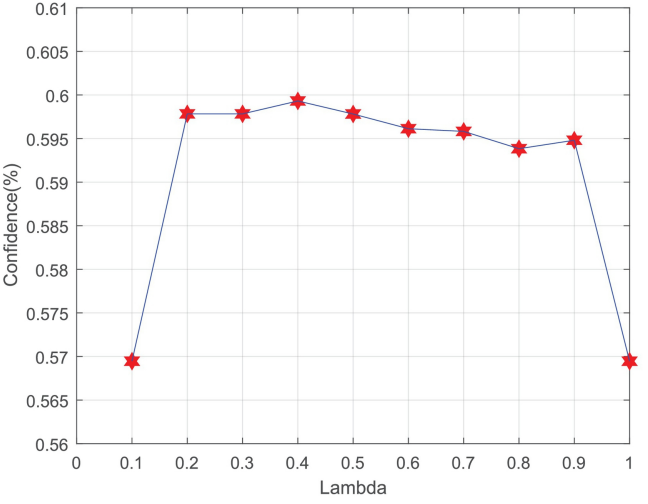
(c) Database

(d) JPA

Figure 1

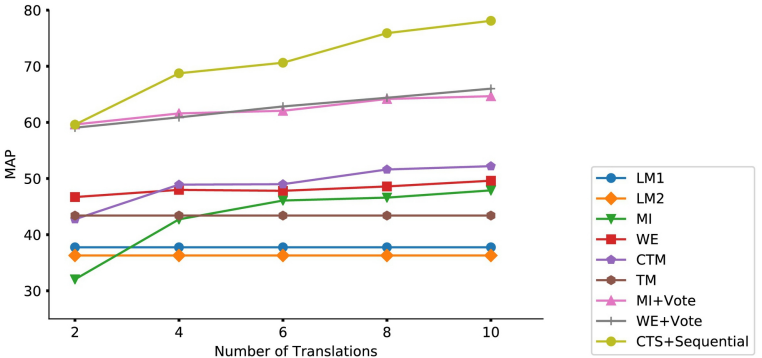


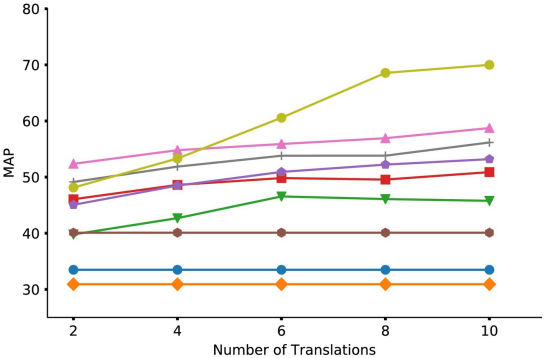
(a) Coverage



(b) Confidence

Figure 2

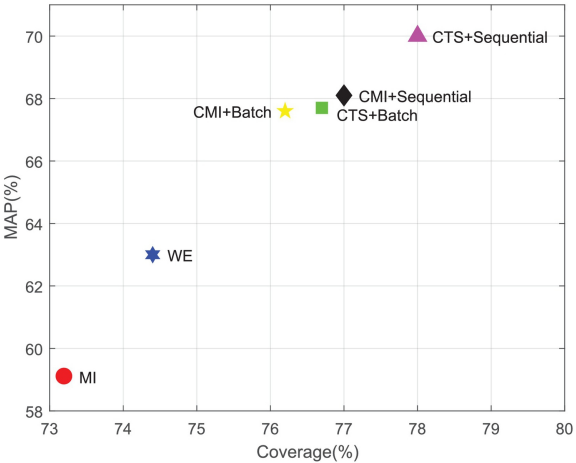




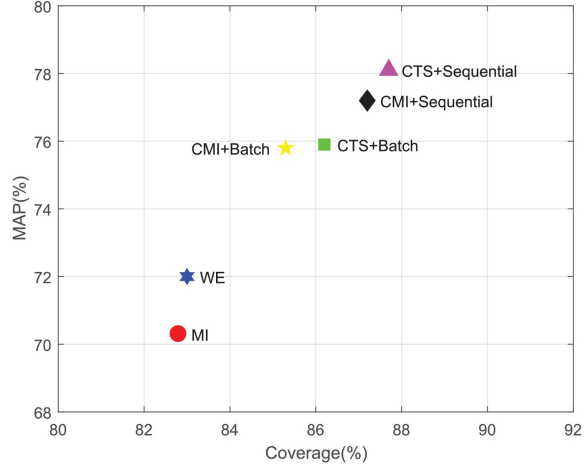
(a) Java

(b) PHP

Figure 3



(a) “PHP”



(b) “JAVA”

Figure 4