Expert Systems With Applications 193 (2022) 116433

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Attention-based skill translation models for expert finding

Zohreh Fallahnejad, Hamid Beigy *

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Keywords:
Expert finding
Semantic matching
Translation models
StackOverflow

ABSTRACT

The growing popularity of community question answering websites can be seen by the growing number of users. Many methods are proposed to identify talented users in these communities, but many of them suffer from vocabulary mismatches. The solution to this problem can be found in translation approaches. The present paper proposes two translation methods for extracting more relevant translations. The proposed methods rely on the attention mechanism. The methods use multi-label classifiers that take each question as input and predict the skills related to the question. Using the attention mechanism, the model is able to focus on specific parts of the given input and predict the correct labels. The ultimate goal of these networks is to predict skills related to questions. Using word attention scores, we can find out how relevant a single word is to a particular skill. As a result of these attention scores, we obtain more relevant translations for each skill. We then use these translations to bridge the lexical gap and improve expert retrieval results. Extensive experiments on two large sub-collections of the StackOverflow dataset demonstrate that the proposed methods outperform the best baseline method by up to 14.11/% MAP improvement.

1. Introduction

Community Question Answering (CQA) websites are popular knowledge sharing platforms that help users meet their information needs. These communities have created a suitable environment where any kind of questions can be answered with people's participation. In recent years, general-purpose CQAs like Quora and Yahoo! Answers and special-purpose CQAs like StackOverflow have attracted numerous users. Users can submit their questions, provide answers for others' questions, express their opinions, and vote on the quality of existing answers. These communities use gamification mechanisms like reputation points and badges to motivate users to increase their participation.

StackOverflow is the most popular CQA site for technical and programming topics, containing about 19 million questions and 29 million answers (StackOverflow, 2020). On average, a new question is posted every 14.3 s (StackOverflow, 2020). Each question is tagged by its questioner with at least one tag. These tags are usually programming languages, tools, or related concepts that indicate the skills and knowledge required to answer the particular question.

Many questions are asked in CQAs every day, but most of them are answered poorly or go unanswered. In addition, a large number of questions are answered only by a limited number of users. In recent years, various studies have been conducted to apply expert finding methods in CQAs to address such problems (Neshati et al., 2017; Shen et al., 2020; Tang, Lu, Gu, et al., 2020; Tang, Lu, Li, et al., 2020). Finding experts is the problem of finding a group of people with appropriate skills and knowledge for a given query (Neshati, Beigy, & Hiemstra, 2014). Identifying potential experts with the goal of nurturing and retaining these users can also increase the quality of these communities (Fu et al., 2016; van Dijk et al., 2015).

The popularity of StackOverflow has made it the largest programming community. This platform has become one of the 50 most popular platforms in the world and has more than 120 million monthly visitors (StackOverflow, 2020). StackOverflow is a great place to post job opportunities and find talented people. This platform can help recruiters find and hire the most talented people for their company. Finding the right people to match job openings was another motivation for studies on finding experts (Nobari et al., 2020; Ramanath et al., 2018). By considering each tag or skill as a query, these studies aim to find individuals with the appropriate knowledge of that skill.

Identifying potential experts for a particular skill in CQA repositories is a difficult task because there is a lexical gap between the skill and the user documents (questions and answers). For example, experts in the Object Oriented Programming (OOP) skill may not use the word "OOP" in their posts, but use words with related concepts such as "class", "object", "public", etc. Traditional methods, such as probabilistic methods (Balog et al., 2009), are based on exact matching and suffer from the vocabulary mismatch problem. To address this problem, researchers have proposed the use of various technique, including topic

* Corresponding author.
E-mail addresses: zfallahnejad@ce.sharif.edu (Z. Fallahnejad), beigy@sharif.edu (H. Beigy).

https://doi.org/10.1016/j.eswa.2021.116433
Received 11 February 2021; Received in revised form 20 November 2021; Accepted 19 December 2021
Available online 11 January 2022

modeling (Momtazi & Naumann, 2013), word embedding (Nobari et al., 2020; Van Gysel et al., 2016), and translation models (Karimzadehgan & Zhai, 2010; Nobari et al., 2020), to capture the semantic relations between words.

Deep learning has achieved significant success in a number of fields such as machine vision, speech recognition, bioinformatics, and natural language processing. In recent advances in natural language processing, CNNs, RNNs, LSTMs, and currently attention mechanisms have been used to extract the semantic information present in the text (Onan, 2019, 2020, 2021; Onan & Toçoğlu, 2021; Yang et al., 2016).

In this paper, we propose two skill translation models for the task of expert finding in CQA. We consider each skill (tag) as a query and deal with ranking experts that have the best knowledge about the given skill. Two attention-based neural networks are proposed for skill translation, which are basically multi-label classifiers that take each question as the input of the network and predict its skill tags. By using the assigned attention scores of each word, the semantic relationship between words and skills is learned, and thus for each skill, a set of the most relevant words is selected as the translation of the skill. These translations can help to overcome the vocabulary mismatches between skills and user documents, resulting in semantic correspondence between them and improvement of expert finding results.

We evaluate the proposed methods on two collections of the StackOverflow dataset. We compare the proposed methods with multiple baselines to predict the best ranking of users for a given skill. We choose two probabilistic models (Balog et al., 2009), the topic modeling-based model (Momtazi & Naumann, 2013), and two translation models (Nobari et al., 2020) as our baselines. The two translation models proposed in (Nobari et al., 2020) are the most similar approaches to the proposed methods. They consider each word individually and compute the probability of translating a particular skill in that word. However, we analyze each word in the context in which it is used. The extensive experiments show the effectiveness of the proposed methods. The experimental results show that the proposed methods improve the MAP measure compared to the baseline methods, even when a small number of translations are used for each skill. We have made the source code and dataset publicly available for further research.[1]

The main contributions of this work can be summarized as follows: (1) proposing two attention-based neural networks for the skill translation task, (2) using word attention scores assigned by the proposed multi-label document classifier models for the purpose of skill translation, and (3) comparing the proposed models with several baseline models on two collections of the StackOverflow dataset.

The rest of this paper is organized as follows. In Section 2, we review related work on this topic. Section 3 describes the baseline methods. In Section 4, the proposed method will be discussed in detail. Experimental results are reported in Section 5. Section 6 will provide a detailed discussion of the proposed method. Finally, we conclude the paper and point out future work in Section 7.

2. Related work

In this section we give an overview of previous research related to our work. First, we give an overview of previous work on expert finding. Then, we focus on reviewing related work on CQAs. Finally, we examine related studies that aim to solve the vocabulary gap challenge.

2.1. Expert finding

Finding experts is a well-defined problem in the *Information Retrieval* (IR) community (Hashemi et al., 2013). Several studies have been conducted to address this problem in bibliographic networks (Fischer et al., 2019; Yang et al., 2020), social networks (Li et al., 2020; Ma et al., 2019; Neshati, Hashemi, & Beigy, 2014; Neshati et al., 2014c), CQAs (Dehghan et al., 2019, 2020; Gharebagh et al., 2018; Neshati et al., 2017), LinkedIn (Ramanath et al., 2018), Ticket Management System (Xu & He, 2018), and Citizen Science Platforms (Saoud & Fontaine, 2018).

Several studies have used probabilistic models for the expert finding task. Balog et al. (2009) formalized the expert finding problem using two probabilistic generative language models, namely a candidate-based model and a document-based model. The candidate-based model generates a language model for each candidate using the documents associated with the candidate, and then estimates the association between query terms and the candidate's language model. The document-based model retrieves the relevant documents for a given query and then ranks the authors of the relevant documents.

There are several studies on graph-based approaches for finding experts. These approaches construct graphs of users and their relations or graphs of users and documents and their associations to represent the main elements of expert finding models. Liu et al. (2020) proposed two graph convolutional neural networks using user profiles, question and answer text information, and social behavior network. Saoud and Fontaine (2018) used user comments and social relations to build a weighted directed graph and proposed a weighted PageRank algorithm to estimate the expertise of each user. Xu and He (2018) proposed an expert recommendation algorithm for ticket routing by building an expert collaboration network from problem descriptions and resolved sequences of historical tickets. These tickets and candidate experts are embedded in a low-dimensional space called expertise space to exploit the semantic similarities between them. Then, a two-stage expert recommendation algorithm was used to find the best expert to solve a new ticket.

2.2. Community Question Answering (CQA)

In recent years, expert search in CQAs has attracted much attention. CQA networks have become valuable resources for searching and sharing knowledge between users.

Every day, many questions are asked in CQAs. Most of these questions receive low quality answers or go unanswered. Several question routing methods have been proposed to recommend newly asked questions to experienced users. Shen et al. (2020) developed an approach to embed heterogeneous information networks for the task of question routing. Wang et al. (2017) proposed convolutional neural networks to predict the best possible answers for newly posed questions. Tang, Lu, Gu, et al. (2020) proposed a new expertise rating system, called ExpertRating (ER), which incorporates user contribution quality and user preferences. They also developed a Gaussian-Gamma mixture clustering model for identifying experts that uses the ER ratings to train users' probability distribution parameters. Tang, Lu, Li, et al. (2020) introduced an attention-based variant of Factorization Machines (FM) called Hierarchical Attentional Factorization Machines (HaFMRank) for expert recommendations using two levels of attention. Kundu et al. (2021) proposed a topic-sensitive hybrid expertise retrieval system that estimates a user's expertise for answering a given question based on three new expertise signatures: Knowledge, Reputation, and Authority.

Considering the dynamic and temporal aspects of expertise is another line of research. Using textual, behavioral, and temporal features, van Dijk et al. (2015) developed a semi-supervised learning method that predicts whether a user will be an expert in the long run. Fu et al. (2016) presented a temporal behavioral analysis of user activities in CQA and analyzed the changes in activity frequency along with response speed. They proposed a Temporal Behavior Model to identify potential experts in the early stages of their participation. Neshati et al. (2017) introduced a new problem of future expert identification: Given expertise in the current time, the goal is to predict the probability of becoming an expert in the future. They proposed two learning methods and studied the effects of four feature sets, namely

[1] https://github.com/zfallahnejad/attention-based-skill-translation.

user behavior, topic similarity, topic transition, and emerging topics. He et al. (2020) considered the dynamics of user interest and expertise for expert recommendations. They proposed an LSTM model to capture the dynamic change in the user's short-term interests. Zhang et al. (2020) considered the temporal dynamics of answering behavior and proposed a temporal context-aware representation for each user.

Finding expert users with a particular shape of expertise is also a new line of research in CQAs. Gharebagh et al. (2018) categorized Stackoverflow users into three groups: C-shaped, T-shaped, and non-experts users. C-shaped users are those who have expertise in more than one skill area, while T-shaped users are those users who have expertise in one skill area and have general knowledge in others. The shape of expertise has a great impact on the cost of recruitment and hiring T-shaped users seems to be a cost-effective approach. Gharebagh et al. (2018) proposed two probabilistic models to find T-shaped expert users with expertise in a particular skill. Dehghan et al. (2019) proposed a neural network that takes a temporal expertise tree for each candidate and predicts the shape of their expertise. Dehghan et al. (2020) also proposed a neural network for the task of T-shaped expert finding. Their proposed network consists of two parallel CNN networks that jointly extract local and position invariant features from the documents of a pair of user and expertise domain to predict the shape of the user's expertise.

2.3. Semantic matching

Vocabulary mismatch between query and document is an important challenge in IR. Traditional methods, such as probabilistic methods, rely on exact matching and suffer from this problem. Semantic matching is an attempt to solve this problem and several studies have addressed it by using methods such as Topic Modeling (Momtazi & Naumann, 2013; Yi & Allan, 2009), and Translation Models (Nobari et al., 2020).

One of the most popular methods of semantic matching is topic modeling. Topic modeling approaches, such as Latent Dirichlet allocation (LDA), discover latent topics in a document collection (Blei et al., 2003). These extracted topics can serve as a bridge to overcome the mismatch between the vocabulary of a query and the documents. Yi and Allan (2009) presents a comparative study of the use of topic modeling for information retrieval purposes. Momtazi and Naumann (2013) employed a topic modeling method for the task of finding experts.

Recently, several deep neural networks have been used to solve this problem and have shown exceptional results. Van Gysel et al. (2016) learned distributed representations for words and candidates using a neural language modeling approach. Liang (2019) proposed an unsupervised semantic generative adversarial networks (USGAN) for expert finding, which consists of a discriminative model and a generative model, and aims to learn semantic representations of experts and words in an unsupervised manner. Zhao et al. (2016) formulated the expert finding problem as the problem of learning a ranking metric to measure the quality of each user for answering a given question. They built a heterogeneous CQA network using users' social relations and users' relative quality scores on given questions. They developed a random walk-based learning system to learn the semantic representation of questions and users along with embedding the ranking metric for questions and users. Cifariello et al. (2019) presented a semantic approach for expert finding called Wikipedia Expertise Ranking (Wiser) that takes advantage of entity linking for semantic matching.

Nobari et al. (2020) proposed two translation approaches for the expert finding task. Their first translation model is a statistical model that translates a given skill (query) into a set of relevant words based on mutual information. Their second translation model is based on a word embedding approach that learns a mapping between the topic space and the skill space. The word embedding approach translates a given skill into a set of relevant words embedded in the skill space near that skill.

3. Baselines

In this section, we will describe several methods for finding experts, which we use as the baselines. We use document-based and candidate-based models proposed in Balog et al. (2009). These models formalize the expert finding problem using probabilistic generative language models. The third baseline is a topic modeling approach for expert finding, proposed in Momtazi and Naumann (2013). The fourth and fifth baselines are a mutual information-based model and a Word Embedding-based model proposed by Nobari et al. (2020). These models translate each skill into a set of relevant words to improve the retrieval performance of expert finding.

3.1. Candidate-based Model (CM)

The candidate-based model is one of the probabilistic generative language models proposed in Balog et al. (2009). The candidate-based model uses the documents associated with the candidates to generate a multinomial language model θ_{ca} for each candidate. Given this generated language model, for each skill sk, the relevance probability of a candidate ca is calculated in the following way:

$$p(sk|ca) = p(sk|\theta_{ca}) = \prod_{t \in sk} p(t|\theta_{ca})^{n(t,sk)}, \tag{1}$$

where the term t is one of the terms representing the skill sk, $p(t|\theta_{ca})$ represents the generation probability of the term t given expert candidate ca, and $n(t, sk)$ indicates the term frequency of t in the skill sk. The probability of term t at expert candidate ca is estimated from all documents related to candidate ca. Then, this probability is smoothed by assigning nonzero probabilities to the unseen terms:

$$p(t|\theta_{ca}) = (1 - \lambda_{ca}) \left(\sum_{d \in D_{ca}} p(t|d, ca) p(d|ca) \right) + \lambda_{ca} p(t), \tag{2}$$

where D_{ca} represents the set of documents associated with candidate ca, λ_{ca} denotes the smoothing parameter, $p(t)$ denotes the probability of term t, and $p(d|ca)$ is the probability that document d was generated by candidate ca. By assuming conditional independence between document d and candidate ca, $p(t|d, ca)$ can be simplified to $p(t|d)$, which represents the probability of term t in document d and can be estimated using maximum likelihood.

3.2. Document-based Model (DM)

The document-based model is another probabilistic generative language model proposed in Balog et al. (2009). The document-based model retrieves the relevant documents for a given skill, and then ranks the authors of the relevant documents as follows:

$$p(sk|ca) = \sum_{d \in D_{ca}} p(sk|d, ca) \dot{p}(d|ca), \tag{3}$$

where $p(d|ca)$ represents the probability that document d is generated by candidate ca and $p(sk|d, ca)$ represents the probability that skill sk is generated from document d and candidate ca. By assuming conditional independence between all terms of skill sk, $p(sk|d, ca)$ can be calculated as follows:

$$p(sk|d, ca) = \prod_{t \in sk} p(t|d, ca)^{n(t,sk)}. \tag{4}$$

By assuming conditional independence between document d and candidate ca, $p(t|d, ca)$ can be simplified as $p(t|\theta_d)$, which denotes the probability of generating term t from the language model of document d. Then, this probability is smoothed using the background collection probability of term t to avoid zero probabilities, so that:

$$p(sk|ca) = \sum_{d \in D_{ca}} \prod_{t \in sk} \left\{ (1 - \lambda_d) p(t|d) + \lambda_d p(t) \right\}^{n(t,sk)} p(d|ca), \tag{5}$$

where λ_d is the smoothing parameter.

3.3. Topic-based Model (TM)

To overcome the vocabulary mismatch between skills and the expert candidates, Momtazi and Naumann (2013) proposed a topic modeling approach. They used LDA to extract the main topics of the document collection. These topics were used as a bridge between skills and expert candidates:

$$p(sk|ca) = \sum_{z \in Z} p(sk|z, ca) \cdot p(z|ca), \tag{6}$$

where z represents a topic from the set of extracted topics Z, $p(sk|z, ca)$ is the probability of generating skill sk given topic z and candidate ca, and $p(z|ca)$ is the probability of generating topic z given candidate ca. By assuming conditional independence between topic z and candidate ca, $p(sk|z, ca)$ simplifies to $p(sk|z)$, which can be estimated by generating all terms in skill as follows:

$$p(sk|z) = \prod_{t \in sk} p(t|z)^{n(t,sk)}. \tag{7}$$

Using Bayes' theorem to compute $p(z|ca)$ and smoothing to avoid zero probabilities of $p(t|z)$, the final estimate of the topic-based model is computed as

$$p(sk|ca) \propto \sum_{z \in Z} \prod_{t \in sk} \left[(1 - \lambda_z) p(t|z) + \lambda_z . p(t) \right]^{n(t,sk)} p(ca|z) p(z), \tag{8}$$

where λ_z is a smoothing parameter and $p(ca)$ is assumed to be a constant value for all candidates. The probability of term t at topic z, $p(t|z)$, and the probability of candidate ca at topic z, $p(ca|z)$, can be computed using the multinomial distributions estimated by LDA. $p(z)$ is the topic prior probability, which can be assumed to be uniform for all topics.

3.4. Mutual Information based Model (MIM)

Nobari et al. (2020) proposed two skill translation approaches. Their first translation model is based on mutual information. Mutual information calculates the statistical dependence between two random variables and measures how much one of these random variables tells about another. In this model, they calculate the mutual information for each pair of skill sk and word w as

$$MI(sk, w) = \sum_{A_{sk}=0,1} \sum_{A_w=0,1} p(A_{sk}, A_w) log \frac{p(A_{sk}, A_w)}{p(A_{sk}) p(A_w)}, \tag{9}$$

where A_{sk} is a binary variable indicating whether an answer is related to skill sk, and A_w is a binary variable indicating whether word w is present in an answer. Normalizing the mutual information scores yields the translation probability for each pair of skill sk and word w is:

$$p(w|sk) = \frac{MI(sk, w)}{\sum_{w'} MI(sk, w')} \tag{10}$$

After computing the MI scores for the candidate words, the top n words with the higher MI score are selected as the translation of skill sk. These translations can be used to estimate $p(sk|ca)$ as.

$$\begin{aligned} p(sk|ca) &= p(w_1, \dots, w_n|ca) \\ &\propto \sum_{d \in D_{ca}} p(w_1, \dots, w_n|d, ca) . p(d|ca) \end{aligned} \tag{11}$$

where D_{ca} represents the set of documents associated with candidate ca. Using Bayes' rule, we can rewrite this equation as.

$$p(sk|ca) \propto \sum_{d \in D_{ca}} p(w_1, \dots, w_n|d, ca) p(ca|d) p(d), \tag{12}$$

where $p(ca)$ is assumed to be the same for all candidates and ignored from Eq. (12). Since each post of StackOverflow belongs to exactly one author, we have $p(ca|d) = 1$. By assuming conditional independence between the candidate and the words, this equation can be simplified as.

$$p(sk|ca) \propto \sum_{d \in D_{ca}} p(w_1, \dots, w_n|d) p(d), \tag{13}$$

where $p(w_1, \dots, w_n|d)$ denotes the probability that the translated words are relevant given the document and $p(d)$ denotes the probability of the document. We can estimate $p(w_1, \dots, w_n|d)$ in the following way:

$$p(w_1, \dots, w_n|d) = \begin{cases} 0, & \text{if } w_1, \dots, w_n \notin d \\ 1, & \text{otherwise} \end{cases} \tag{14}$$

where $p(w_1, \dots, w_n|d)$ is equal to 1 if the document d contains at least one of these translation words, otherwise it is equal to zero.

Most of the traditional expert finding models assume a uniform value for $p(d)$ and ignore it in the final ranking. Nobari et al. (2020) proposed a new document quality score called Voteshare measure to estimate $p(d)$. Voteshare is a measure of the quality of a single document compared to other documents within a question thread. Voteshare of document d is calculated by dividing its vote score by the sum of the vote scores of all documents in the thread in which document d appears. A document's vote score is the difference between the number of upvotes and downvotes given by users.

3.5. Word Embedding based Model (WEM)

Nobari et al. (2020) proposed a different translation approach to the expert finding task, which is a word embedding based method. To reduce the vocabulary gap between terms and skills, they proposed to embed skills and document terms into a new space called skill domain space. The mapping function of each word and skill into the skill-area space was developed as follows.

$$p_{we}(sk|w) = \frac{1}{z} e^{W_{LDA} W_C + b}, \tag{15}$$

where W_{LDA} is a T-dimensional topic space representation of each word in LDA space (T denotes the number of extracted topics using LDA), W_C is a $T \times S$ projection matrix mapping the topic representation of each word to its representation in skill domain space (S is the number of skills), and b denotes an S-dimensional bias vector. The result of this mapping function must be normalized by $z = \sum_{sk'} e^{W_{LDA} \cdot W_C + b}$ to be normalized, which is the sum of the translation probability of word w to each possible skill sk'.

Eq. (15) defines a neural network with a single hidden layer. The matrix W_C and the vector b are the parameters of this network. These parameters are estimated using error backpropagation. During training, the observed probability of translating each document term to a given skill is defined as follows:

$$p_{observed}(sk|w) = \frac{tf(sk, w)}{tf(w)}, \tag{16}$$

where $tf(sk, w)$ denotes the number of occurrences of the term w in the answers associated with the skill sk, and $tf(w)$ represents the total number of occurrences of the term w in the training set. In training, the loss function for a single batch of m instances is defined as follows:

$$L(W_C, b) = \frac{1}{m} \sum_{i=1}^{m} H(p_{we}, p_{observed}) + \frac{\lambda}{2m} \left(\sum_{i,j} W_{C_{i,j}}^2 \right), \tag{17}$$

where $H(p_{we}, p_{observed})$ is the cross entropy of p_{we} and $p_{observed}$ and λ is the weight regularization parameter.

After the training phase, the probability $p(sk|w)$ for each skill sk and vocabulary word w can be estimated using Eq. (15). By applying Bayes' theorem, we have $p(w|sk) \propto p(sk|w) \cdot p(w)$ for each skill sk, where $p(w)$ is the probability of word w estimated using the TF–IDF value. Using $p(w|sk)$, the top n words with the higher translation probability are selected as the translation of skill sk. These translations can be used to estimate the probability $p(sk|ca)$ using Eqs. (11)–(13).

4. The proposed models

Searching for experts on CQA websites such as StackOverflow addresses the problem of finding and retrieving an ordered list of experts who have relevant knowledge and expertise in a given skill domain. Given a skill sk, the task is to retrieve a ranked list of candidates that are likely to be an expert. Formally, for a given skill sk, we want to estimate $p(ca|sk)$ for each candidate ca and then rank them according to this probability. We can estimate $p(ca|sk)$ by applying Bayes' rule:

$$p(ca|sk) = \frac{p(sk|ca)p(ca)}{p(sk)}. \tag{18}$$

Given a constant value for $p(ca)$ and $p(sk)$, we can use $p(sk|ca)$ as the expertise score of candidate ca. Traditional methods mostly use exact matching approaches that are not sufficient to estimate the relationship between skills and candidate documents.

The skills and user documents may use different words for a given concept or use different words in the concept hierarchy. Therefore, there is a lexical gap between skills and user documents, making it difficult to find experts. Exact matches used in traditional methods do not take these lexical gaps into account. A translation model that converts each skill into related words can reduce the vocabulary gap problem and provide a more accurate estimation of relevancy.

These translations can be used for query expansion. With the help of extracted translations, we can identify expert candidates who have documents related to the given skill (query) or its translations.

For each skill sk, we want to find the n most related words to sk. Let $tr_1, \dots, tr_n$ represent the top n translations of sk. Similar to the Eqs. (11) and (12), for each candidate ca we use these translations to estimate the probability $p(sk|ca)$:

$$\begin{aligned} p(sk|ca) &= p(tr_1, \dots, tr_n|ca) \\ &\propto \sum_{d \in D_{ca}} p(tr_1, \dots, tr_n|d, ca).p(d|ca) \\ &\propto \sum_{d \in D_{ca}} p(tr_1, \dots, tr_n|d).\frac{p(ca|d).p(d)}{p(ca)} \\ &\propto \sum_{d \in D_{ca}} p(tr_1, \dots, tr_n|d).p(d) \end{aligned} \tag{19}$$

where D_{ca} represents the set of responses associated with candidate ca. We set $p(tr_1, \dots, tr_n|d)$ to 1 if the document d contains at least one of these translation words, otherwise $p(tr_1, \dots, tr_n|d) = 0$. Finally, we can use the binary and Voteshare scores to estimate the document probability $p(d)$, where the binary score assumes $p(d) = 1$ and the Voteshare score estimates $p(d)$ using the Voteshare measure proposed in Nobari et al. (2020).

In this paper, for each skill, we want to find the most related translations. We propose two attention-based multi-label classification networks for the skill translation task. The words of a question are fed into these networks, and these networks predict the question-related skills as the output of the last layer. We used the attention mechanism in the design of these networks, which allows the model to focus on specific parts of the input to correctly predict the document labels (skills). Although the ultimate goal of these networks is to predict document skills, we want to take advantage of word attention scores (middle layer weights) for the skill translation task. With these two networks, we want to know how relevant a single word is to each skill so that we can find the most relevant translation words for each skill. The proposed networks consist of the following layers: Word embedding layer, Bidirectional LSTM layer, Attention Layer, and Dense Layer. Unlike two translation approaches proposed in Nobari et al. (2020) that consider each word in isolation, the proposed models use context information for each word.

In the following subsections, we first describe in detail the design of the proposed attention-based multi-label classification networks. We then explain how the learned models can be used to obtain semantic translations for each skill.

4.1. Attention-based Skill Translation Model 1 (ASTM-1)

In this section, we propose the first attention-based skill translation model, which we call the *ASTM-1 model*. Fig. 1 shows the architecture of the proposed ASTM-1 model. It consists of the following layers: Word embedding layer, bidirectional LSTM layer, attention layer and dense layer. In the following, we describe each layer.

Word embedding. Similar to the method given in (Dehghan & Abin, 2019), we applied the Latent Dirichlet Allocation (LDA) algorithm to a given set of StackOverflow post collection to learn a T-dimensional topic representation for each word in our vocabulary. We use these topic representations as the initial values of the word embeddings and will continue to learn the word embeddings during the training phase of the network.

Suppose that the training data is given as N pairs of questions and skill vectors $\{(q_i, sk_i)\}_{i=1}^{N}$, where q_i represents the ith question in a given StackOverflow collection, $sk_i \ in \{0,1\}^S$ represents an S-dimensional binary vector representing relevant skills of q_i and S is the number of skills. This model accepts a word sequence $w_{i1}, \dots, w_{iL}$ for the question q_i of the training data, where L is the maximum length of the input questions. In this layer, each word is converted into a T-dimensional topic representation. Let $x_{ij} \in \mathbb{R}^T$ be the T-dimensional word embedding corresponding to w_{ij}, where w_{ij} is the jth word in the question q_i. We represent the question q_i by a sequence of its constituent word embeddings as $q_i = [x_{i1}, \dots, x_{iL}]$ to use as input to the subsequent layer, where x_{ij} is initialized by the topic representation of the jth word in the question q_i.

Bidirectional LSTM (Bi-LSTM). A bidirectional LSTM (Bi-LSTM) layer is trained on the input sequences in both forward and backward directions to capture the past contextual information and the future information. Given an input question $q_i = [x_{i1}, \dots, x_{iL}]$, the Bi-LSTM returns a hidden state h_{ij} at time step j using the following equations:

$$\begin{aligned} \vec{h}_{ij} &= \overrightarrow{LSTM}(x_{ij}, h_{i(j-1)}) \\ \overleftarrow{h}_{ij} &= \overleftarrow{LSTM}(x_{ij}, h_{i(j-1)}) \\ h_{ij} &= [\vec{h}_{ij} \parallel \overleftarrow{h}_{ij}] \end{aligned} \tag{20}$$

where x_{ij} is the word embedding of the jth word in the question q_i and $h_{i(j-1)}$ is the generated hidden state of time step $j-1$. For time step j, the forward LSTM and the backward LSTM compute the forward hidden state $\vec{h}_{ij}$ and the backward hidden state $\overleftarrow{h}_{ij}$, respectively. We concatenate the forward and backward hidden states with the concatenation operator $\parallel$ to produce the hidden state h_{ij}. The hidden state h_{ij} encodes information about the jth word and its surrounding context words, so h_{ij} can be viewed as a context-aware word embedding for the jth word of the input question. A dropout is then applied to the sequence of hidden states and the result is used as the input of the subsequent layer.

Attention layer. The contextual word embedding learned by the Bi-LSTM layer is passed on to the attention layer. First, we apply a single-layer multilayer perceptron (MLP) to each hidden state to obtain u_{ij} as a projection of h_{ij} into the attention space:

$$u_{ij} = tanh(W_w h_{ij} + b_w). \tag{21}$$

We use the attentional approach to measure the semantic connection between words and skills. Similar to the method given in Yang et al. (2016), we measure the importance of the jth word in the question q_i according to the following equation:

$$\alpha_{ij} = \frac{exp(u_{ij}^{\top} c)}{\sum_j exp(u_{ij}^{\top} c)}, \tag{22}$$

where c is a trainable context vector. Finally, we use the word attention scores to obtain the question representation as follows:

$$v_i = \sum_{j=1}^{L} \alpha_{ij} h_{ij}, \tag{23}$$

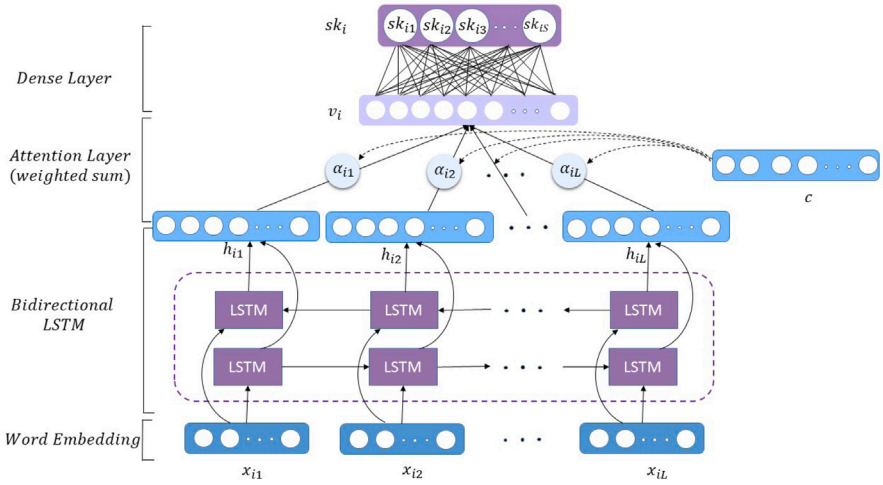

Fig. 1. Schematic representation of Attention-based Skill Translation Model 1.

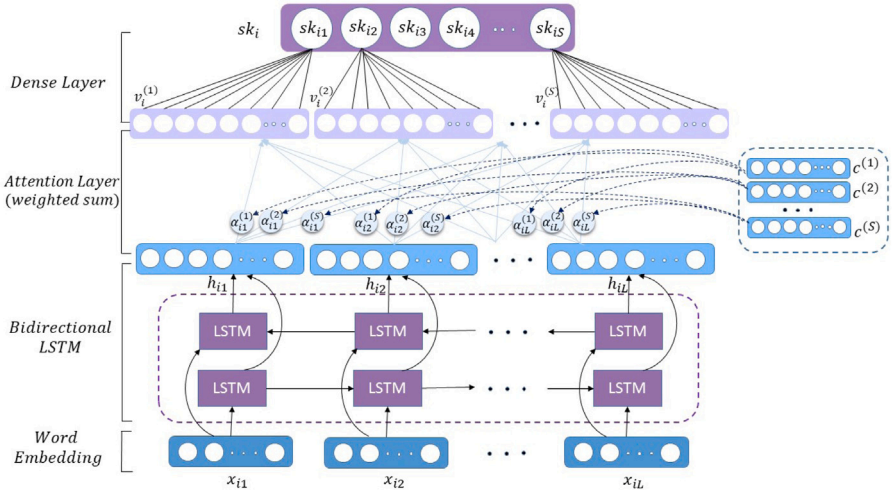

Fig. 2. Schematic representation of Attention-based Skill Translation Model 2.

where v_i represents a weighted sum of the word hidden states based on the attentional scores. Then, a dropout is applied to the question representation and the result is used as the input of the subsequent dense layer.

Dense layer. The dense layer is a fully-connected layer with sigmoidal activation function. We applied the sigmoidal activation function to generate the probability of assigning each skill label for the input question.

$$\hat{sk}_i = sigmoid(W_d v_i + b_d). \tag{24}$$

The final prediction $\hat{sk}_i = (\hat{sk}_{i1}, \dots, \hat{sk}_{iS})$ is a binary vector in which $\hat{sk}_{ik}$ represents the probability that the question q_i is relevant to the skill sk_k. We use the binary cross-entropy loss over the sigmoid output to train the proposed network:

$$\min_{\Theta} -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{S}[sk_{ik} log(\hat{sk}_{ik}) + (1 - sk_{ik}) log(\hat{sk}_{ik})], \tag{25}$$

where m denotes the number of questions in a single batch, Θ represents the parameters of this network, $\hat{sk}_{ik}$ is the model prediction for instance i on skill k, and sk_{ik} is the ground truth value. The ground truth value is $sk_{ik} = 1$ if the ith question is tagged with the kth skill, otherwise $sk_{ik} = 0$.

After training the network, we now focus on finding the relevant translations for each skill. Let the test data be given as N' pairs of questions and skill vectors $\{(q_i, sk_i)\}_{i=1}^{N'}$, where q_i represents the ith question in the test data and $sk_i in \{0, 1\}^S$ represents an S-dimensional binary vector representing relevant skills of q_i. We feed each question in the test data to obtain the attention scores associated with its words. We define a relevance score to measure the semantic relatedness between each word w in the vocabulary and the kth skill as follows:

$$Score(w, sk_k) = \sum_{i=1}^{N'} \mathbb{1}_{sk_{ik}=1} \sum_{j=1}^{L} \mathbb{1}_{w_{ij}=w} \alpha_{ij}, \tag{26}$$

where sk_{ik} indicates the relevance or irrelevance of sk_k in labeling q_i, w_{ij} represents the jth word in question q_i, and $\mathbb{1}_{\cdot}$ is an indicator function. In other words, the relevance score of the word w and the kth skill is defined as the sum of the attentional scores associated with each occurrence of the word w in a question labeled with the kth skill.

For each skill sk, we sort the relevance score of all words in descending order and select the top n words as translations of the sk skill (in our experiments, we select $n = 10$ top related words). Finally, we can estimate the probability $p(sk|ca)$ for each candidate ca using Eq. (19).

4.2. Attention-based Skill Translation Model 2 (ASTM-2)

In this section, we propose the second attention-based skill translation model, which we call the *ASTM-2 model*. Fig. 2 shows the

architecture of the proposed ASTM-1 model. It consists of the following layers: Word embedding layer, bidirectional LSTM layer, attention layer and dense layer. The first two layers of ASTM-2 are similar to ASTM-1, so we only describe two other layers in the following.

Attention layer. In the attention layer of the ASTM-1 model, we used a single context vector, which can be considered as a representation of an ideal and informative word. Using a single context vector can be a big limitation for our model, so in the second model, we propose to use a different context vector for each skill. Similar to the first model, we first apply a single layer multilayer Perceptron (MLP) to each hidden state. Then, we use a new attentional approach to measure the semantic connection between words and skills as follows:

$$u_{ij} = tanh(W_w h_{ij} + b_w) \tag{27}$$

$$\alpha_{ij}^{(k)} = \frac{exp(u_{ij}^{\top} c^{(k)})}{\sum_j exp(u_{ij}^{\top} c^{(k)})}, \tag{28}$$

where c^k is a trainable context vector for the kth skill and $\alpha_{ij}^{(k)}$ is the attentional score of the jth word in the question q_i according to the kth skill. Finally, we use the word attention scores to obtain the question representation according to each skill as follows.

$$v_i^{(k)} = \sum_{j=1}^{L} \alpha_{ij}^{(k)} h_{ij}, \tag{29}$$

where $v_i^{(k)}$ represents a weighted sum of the word hidden states based on the attentional scores for the kth skill. A dropout is then applied to this question representation and the result is used as the input of the subsequent dense layer.

Dense layer. The output vector from the attention layer is then passed through a fully connected sigmoid layer, which outputs a posterior probability for each skill given a q_i question:

$$\hat{sk}_{ik} = sigmoid(W_d^{(k)} v_i^{(k)} + b_d^{(k)}), \tag{30}$$

where $W_d^{(k)}$ and $b_d^{(k)}$ are trainable parameters for the kth skill in the output layer, respectively.

Similar to ASTM-1, we use the binary cross-entropy loss over the sigmoid output to train our network. After training, we use the trained model to estimate the importance of each word in the context of each skill. Let the test data be given as N' pairs of questions and skill vectors $\{(q_i, sk_i)\}_{i=1}^{N'}$, where q_i represents the ith question in the test data and $sk_i\ in \{0,1\}^S$ represents an S-dimensional binary vector representing the relevant skills of q_i. We feed each question in the test data to obtain the attention scores associated with its words. We defined a relevance score to measure the semantic relatedness between each word w in the vocabulary and the kth skill as follows:

$$Score(w, sk_k) = \sum_{i=1}^{N'} \mathbb{1}_{sk_{ik}=1} \sum_{j=1}^{L} \mathbb{1}_{w_{ij}=w} \alpha_{ij}^{(k)} \tag{31}$$

where sk_{ik} indicates the relevance or irrelevance of sk_k in labeling q_i, w_{ij} represents the jth word in question q_i, $\alpha_{ij}^{(k)}$ represents the attentional score of the jth word in question q_i for the kth skill, and $\mathbb{1}_{\cdot}$ is an indicator function. In other words, the relevance score of the word w and the kth skill are defined as the sum of the attentional scores associated with each occurrence of the word w in a question marked with the kth skill.

For each skill sk, we sort the relevance score of all words in descending order and select the top n words as translations of the skill sk. Finally, we can estimate the probability $p(sk|ca)$ for each candidate ca using Eq. (19).

5. Experiment results

The purpose of this section is to evaluate the performance of the proposed models. We evaluate the performance of these models on some standard datasets and compare the proposed models with some state-of-the-art models with a series of experiments.

5.1. Data collection

We used a snapshot of the StackOverflow dump, which contains over 24 million posts from August 2008 to March 2015. We validate the proposed models on two collections of the original data: the Java collection and the PHP collection. The Java collection contains all questions tagged with "java" and their corresponding answers, which include 810071 questions and 1510812 answers. The PHP collection contains all questions tagged with "PHP" and the corresponding answers, comprising 714476 questions and 1298107 answers.

We select the top 100 most frequent tags from these two collections to use as queries for experts (skills). Relevant experts for each query or skill are determined based on the number of accepted answers and the acceptance ratio of each candidate. Similar to Nobari et al. (2020), we mark users as experts for a skill if they provide an accepted answer to at least 10 questions about that skill and the acceptance ratio of their answers is more than 40%. Due to the smaller size of the PHP collection compared to the Java collection, the threshold for accepted answers is set to 7.

We use questions to train the proposed networks. We select questions with at least one of the top tags (skills) from each collection. To ensure that the quality of our data remains high, we keep only the questions with non-negative score. Thus, we prepared 533426 Java questions and 506305 PHP questions to train our networks. We split the data into three subsets: Training (70%), Validation (10%), and Testing (20%), each with a similar distribution of skills.

5.2. Experimental results

An extensive series of experiments are designed to address the answer to the following research questions in the proposed models:

RQ1 Do the proposed models outperform the state-of-the-art baseline models?

RQ2 If we use Voteshare score as scoring measure for documents, what is the performance of the proposed models?

RQ3 How many translations for each skill can lead to better results?

RQ4 What is the impact of the word embedding dimension on the performance of the proposed models?

RQ5 What is the impact of the collection size on the performance of the proposed models?

RQ6 What translations did the proposed models suggest for each skill?

RQ7 How effective are the proposed models for document classification?

5.2.1. Comparison with baselines (RQ1)

We evaluated the performance of the proposed models using the following metrics: *Mean Average Precision* (MAP), *Precision at the first rank* (P@1), P@5, and P@10. The evaluation results for both Java and PHP data collections are summarized in Table 1. In addition, the standard deviation of the performance metrics is presented in Table 2. Note that these results were computed considering the binary scoring for document likelihood in Eq. (19). According to this table, the performance of our models is significantly better than that of the CM and DM models, which are based on exact matching. Both proposed models perform better than the TM model, which uses topic modeling to solve the lexical gap problem.

The main competitors are MIM and WEM, which have used two skill translation approaches to overcome the vocabulary mismatch problem and improve the expert finding results, and between these two models WEM has achieved the better results. Unlike these models that consider

Table 1
Comparison of the proposed models with the baseline models. The reported results were based on the use of binary scoring.

	Model	MAP	P@1	P@5	P@10
Java	CM	0.377	0.560	0.500	0.440
	DM	0.362	0.540	0.482	0.425
	TM	0.434	0.550	0.530	0.488
	MIM	0.478	0.660	0.604	0.529
	WEM	0.496	0.650	0.626	0.540
	ASTM-1	0.558	0.660	**0.646**	0.563
	ASTM-2	**0.566**	**0.680**	0.640	**0.571**
PHP	CM	0.335	0.570	0.524	0.479
	DM	0.309	0.520	0.478	0.437
	TM	0.401	0.530	0.550	0.491
	MIM	0.458	0.590	0.612	0.561
	WEM	0.509	0.600	0.626	0.581
	ASTM-1	0.533	0.600	0.630	**0.592**
	ASTM-2	**0.551**	**0.640**	**0.638**	0.590

Table 2
Standard deviation of the performance metrics. The reported results were based on the use of binary scoring.

	Model	MAP	P@1	P@5	P@10
Java	ASTM-1	0.179	0.476	0.238	0.208
	ASTM-2	0.174	0.469	0.251	0.209
PHP	ASTM-1	0.187	0.492	0.260	0.240
	ASTM-2	0.191	0.482	0.251	0.250

Table 3
Comparison of the proposed models with the baseline models. The reported results are based on the use of Voteshare scoring.

	Model	MAP	P@1	P@5	P@10
Java	CM	0.449	0.720	0.586	0.514
	DM	0.432	0.690	0.568	0.496
	TM	0.434	0.550	0.530	0.488
	MIM	0.647	0.850	0.736	0.652
	WEM	0.660	**0.860**	0.728	0.661
	ASTM-1	0.652	0.830	**0.738**	0.646
	ASTM-2	**0.666**	0.810	0.728	**0.667**
PHP	CM	0.361	0.620	0.582	0.505
	DM	0.334	0.560	0.540	0.465
	TM	0.401	0.530	0.550	0.491
	MIM	0.587	**0.750**	**0.726**	0.642
	WEM	0.562	**0.750**	0.696	0.621
	ASTM-1	0.578	0.710	0.700	0.639
	ASTM-2	**0.602**	0.730	0.720	**0.646**

Table 4
Standard deviation of the performance metrics. The reported results were based on the use of Voteshare scoring.

	Model	MAP	P@1	P@5	P@10
Java	ASTM-1	0.180	0.378	0.225	0.222
	ASTM-2	0.174	0.394	0.245	0.215
PHP	ASTM-1	0.199	0.456	0.247	0.259
	ASTM-2	0.192	0.446	0.241	0.258

each word individually, our models use the contextual information of each word. Table 1 shows that ASTM-1 model has obtained better results than WEM model which is the best baseline method. Compared to the WEM model, the ASTM-1 model has achieved a relative improvement of 12.5% at MAP in the Java collection and 4.7% at MAP in the PHP collection. Compared to the ASTM-1 model, the ASTM-2 model has achieved better results. Compared to the WEM model, the ASTM-2 model has achieved a relative improvement of 14.1% at MAP in the Java collection and 8.2% at MAP in the PHP collection. Note that the improvement with ASTM-2 is larger compared to ASTM-1. Since ASTM-2 uses different attention scores for each skill, it benefits from a more relevant translation for each skill.

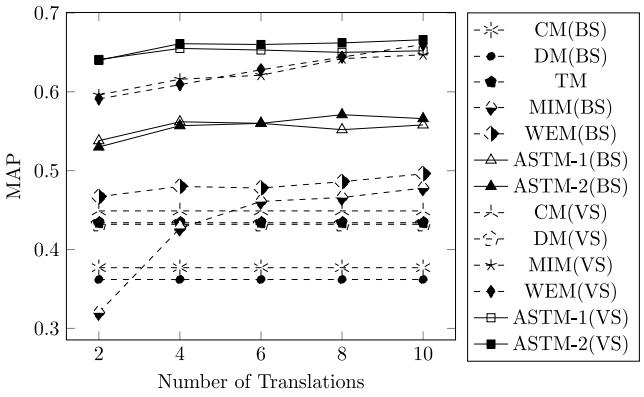

Fig. 3. The effect of the number of translations on MAP metric in Java collection.

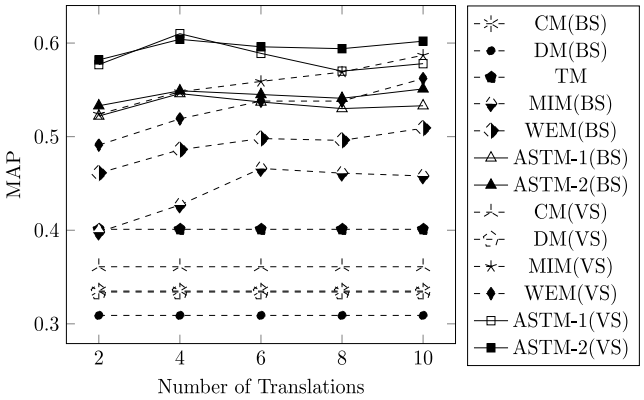

Fig. 4. The effect of the number of translations on MAP metric in PHP collection.

5.2.2. Effect of Voteshare scoring (RQ2)

The results of using the Voteshare scoring measure in the calculation of the proposed models and the baseline models are given in Table 3. Table 3 shows that the proposed models were surprisingly better than the models CM, DM and TM. In addition, the standard deviation of the performance metrics is presented in Table 4. Table 3 shows that the use of the Voteshare scoring method, except in the TM method, improved the results over binary scoring. The TM is a topic-based method, and the probability $p(d)$ does not affect the calculation of Eq. (8), so changing the scoring approach from binary to Voteshare does not change its results. On the Java collection, the ASTM-1 model performed better than the MIM model in terms of MAP, but not better than the WEM model, and was very close to the best model in terms of the P@1 and P@5 criteria. In addition, the ASTM-1 model in the PHP collection was better than the WEM model in terms of MAP, but not better than the MIM model. In both collections, the ASTM-2 model performed better in terms of MAP and P@10, and was very close to the best model in terms of P@1 and P@5.

5.2.3. Effect of number of translations (RQ3)

In this subsection, we will analyze the influence of the number of translations on MAP performance. Figs. 3 and 4 show the impact of the number of translations on the MAP measure for Java and PHP collections, respectively. In these figures, BS stands for Binary Scoring and VS stands for Voteshare Scoring. For the purpose of comprehensive comparison, the results of CM (BS), DM (BS), CM (VS), DM (VS), TM models are also included. As the number of translations increases, the proposed models show consistent and mostly increasing performance. The ASTM-1(BS) and ASTM-2(BS) methods perform better than CM (BS), DM (BS), CM (VS), DM (VS), TM, MIM (BS), WEM (BS) and even

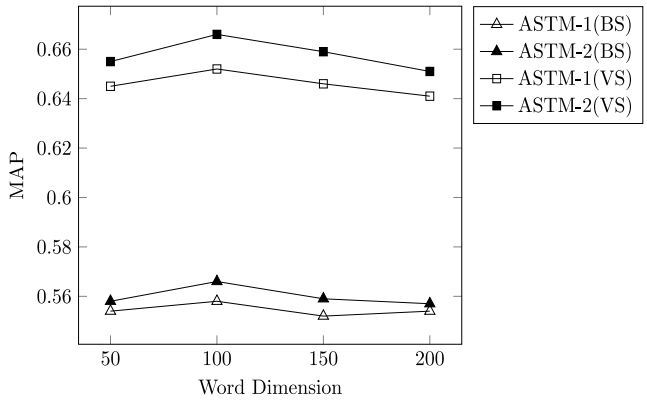

Fig. 5. The effect of word embedding size on the MAP measure in the Java collection.

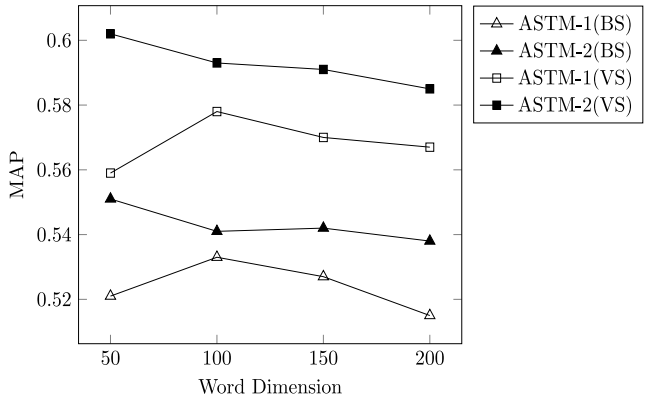

Fig. 6. The effect of word embedding size on MAP measure in PHP collection.

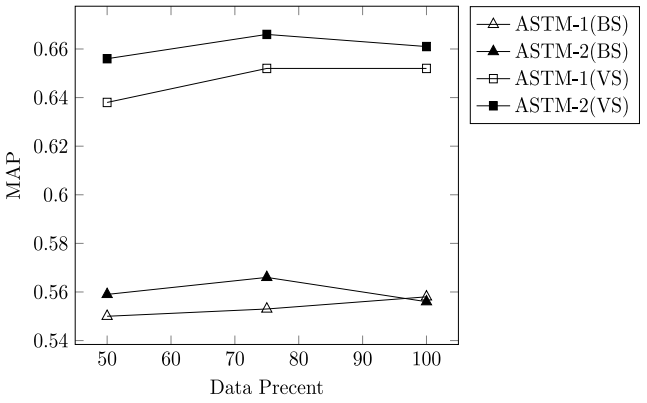

Fig. 7. Effect of collection size on MAP measure in Java collection.

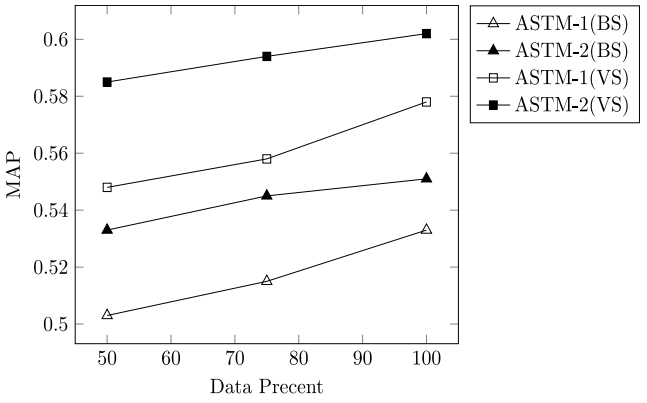

Fig. 8. Effect of collection size on MAP measure in PHP collection.

sometimes MIM (VS) and WEM (VS). ASTM-2(VS) has the best performance compared to its competitors. ASTM-1(VS) also outperforms its competitors in most of the cases. The remarkable performance of the proposed models in the case of using a small number of translations indicates the high quality of the top translations compared to other methods.

5.2.4. Effect of word embedding dimension (RQ4)

In this subsection, we investigate the influence of the dimension of word embedding on the MAP measure. In this experiment, we trained models with 4 different word embedding dimensions: 50, 100, 150 and 200. Figs. 5 and 6 show the influence of word embedding dimension on the performance of MAP for Java and PHP collections, respectively. Changing the word embedding dimension does not significantly affect the performance of the proposed models. Fig. 5 shows that the ASTM-1 and ASTM-2 models perform best for a Java collection when they use 100-dimensional word vectors. Fig. 6 also shows that the ASTM-1 and ASTM-2 models perform best for PHP collection when the word vector dimensions are 100 and 50, respectively.

5.2.5. Impact of collection size (RQ5)

As mentioned in Section 5.1, we prepared 533426 Java questions and 506305 PHP questions to train our networks. In this subsection, we study the effect of collection size on the performance of the proposed models with the MAP measure. In this experiment, we randomly select 50%, 75% and 100% of the samples in each collection. In other words, we prepared 266713, 400069 and 533426 Java questions and 253152, 379728 and 506305 PHP questions to conduct our experiment. Then we divided each of them into three subsets of training (70%), validation (10%) and testing (20%), each with a similar distribution of skills, and used them to train our networks. Finally, the results of this experiment are shown in Figs. 7 and 8. Fig. 7 shows that using 75% of the Java samples is sufficient to obtain a good result. Fig. 8 shows that the best MAP results are obtained when all PHP samples are used.

5.2.6. Suggested translations for skills (RQ6)

Table 5 contains suggested translations for some examples of skills. Due to space limitations, only the top 5 translations are presented.[2] The suggested translations for each skill have been able to describe that skill from different aspects. As the table shows, the ASTM-1 model selects general terms that give us a general concept of the skills; however, the ASTM-2 model provides a more detailed and semantic translation for each skill.

5.2.7. Document classification results (RQ7)

Although the proposed networks are document classifiers, we used attention scores to solve the skill translation problem. In this subsection, we investigate the performance of these networks to solve the multi-label classification problem. We compared their performance using the metrics of macro/micro precision (PA/PI), macro/micro recall (RA /RI), and macro/micro F1 (F1A/F1I). The evaluation results for both Java and PHP data collections are summarized in Table 6.

6. Discussion

Traditional probabilistic methods, such as CM and DM, rely on exact matching and suffer from the problem of vocabulary mismatch. Topic-based methods such as TM attempt to solve this problem by using the

[2] A complete list of skills along with their translations is made publicly available at http://ce.sharif.edu/~zfallahnejad/skill-translations.xlsx.

Table 5
Sample skill translations.

	Skill	Method	tr_1	tr_2	tr_3	tr_4	tr_5
Java	hibernate	ASTM-1	hibernate	entity	query	spring	jpa
		ASTM-2	hibernate	session	jpa	sessionfactory	hql
	selenium	ASTM-1	selenium	webdriver	webelement	test	driver
		ASTM-2	selenium	webdriver	webelement	test	element
	swing	ASTM-1	jframe	jpanel	swing	jtable	gui
		ASTM-2	jframe	swing	jpanel	jtable	jbutton
	arrays	ASTM-1	array	string	arrays	int	arraylist
		ASTM-2	array	arrays	arraylist	for(int	list
PHP	.htaccess	ASTM-1	rewriterule	rewriteengine	htaccess	rewritecond	redirect
		ASTM-2	rewriterule	rewriteengine	htaccess	rewritecond	url
	gd	ASTM-1	image	function	library	images	header('content-type
		ASTM-2	image	images	png	transparency	font
	soap	ASTM-1	soap	wsdl	xml	soapclient	soapfault
		ASTM-2	soap	wsdl	soapclient	soapfault	webservice
	wordpress	ASTM-1	wordpress	plugin	echo	theme	function
		ASTM-2	wordpress	plugin	post	theme	woocommerce

Table 6
The results of the evaluation of the proposed models for the multi-label classification task.

	Method	PA	PI	RA	RI	F1A	F1I
Java	ASTM-1	72.1	80.0	44.2	55.5	52.5	65.5
	ASTM-2	69.1	77.5	46.1	57.1	51.7	65.8
PHP	ASTM-1	66.5	77.1	37.1	47.3	44.6	58.7
	ASTM-2	60.3	74.2	37.8	49.7	42.5	59.5

LDA approach to extract latent topics from the document collection. For topic inference, word co-occurrence information was used, which is a different concept of word similarity, and semantic similarity is less considered. Then, translation methods were presented to translate each skill into a set of relevant words to solve the vocabulary mismatch problem. However, translation methods, such as MIM, also used co-occurrence information. In contrast, translation methods, such as WEM, attempted to capture semantic similarity by considering the proximity of skills and words in a new vector space. The main drawback of this method is that each word is considered individually and contextual information is ignored.

To address the above issues, we proposed an approach that uses contextual information to consider semantic similarity in providing translations. We proposed two attention-based multi-label classification networks. Although the ultimate goal of these networks is to predict question-related skills, we used the attention mechanism to know how much a single word is relevant to a given skill so that we can find the most relevant translation words for each skill.

Expertise generally does not have a precise definition. Depending on the scope of the problem, different definitions of expertise are provided. For example, according to Section 5.1, experts in a community question answering can be defined as users who have at least 10 accepted answers and an acceptance rate of at least 40%. Since the number of these expert users is small compared to the number of questions, their participation will not significantly reduce the number of unanswered questions and low-quality answers. On the other hand, these people are not the only expert users. Consider a group of users who are merely questioning and unwilling to participate in answering. Consider another group of users who have just joined the community and have not been active in the community yet. These two groups of users are not considered experts by this definition, but in fact, there may be expert users among them. Finding professionals among these users and encouraging them to be more active can enhance the quality of the community.

Our solution to this problem is to search for an alternative method to identify potential experts and compare it to other methods, using the experts derived from the definition provided in Section 5.1. The proposed method uses the documents themselves to translate skills into a set of related concepts. Then, the given skill (query) expands to include these related translations. With the help of them, we can identify expert candidates who have documents related to the given skill or its translations, no matter how many accepted answers they have.

According to the experimental results, the proposed approach extracted the semantically related translations, which led to its superiority. The proposed models performed significantly better than the models CM and DM, which shows that our models could achieve better results by providing a solution to overcome the vocabulary gap. Both proposed models performed better compared to the TM model, indicating their efficiency in using contextual information to reduce the vocabulary gap compared to this topic-based model. The superiority of semantic similarity over co-occurrence based similarity leads to better results compared to MIM model. Finally, a key advantage of the proposed approach over WEM is the incorporation of contextual information in providing high-quality translations. The remarkable performance of the proposed models, especially when using a small number of translations, confirms the high quality of the top translations and the usefulness of the proposed approach.

The experimental results show that the ASTM-2 model performs better than the ASTM-1 model. Using a different context vector for each skill in the ASTM-2 resulted in more relevant translations. The results show that the ASTM-1 model selects general terms that give us a general concept of the skills; however, the ASTM-2 model provides a more detailed and semantic translation for each skill.

However, the proposed approach has two drawbacks as well. Our approach only provides words as translations, which is a drawback. To improve the translation of skills, it is better to extract not only words but also related phrases. Additionally, the proposed networks are multi-label classifiers, and their last layer has the same dimensions as the possible document labels. There will be limitations for these networks if the number of labels increases.

7. Conclusions

In this paper, we proposed two attention-based skill translation models to learn the semantic relationship between words and skills. The skill translations were used to solve the vocabulary mismatch problem, resulting in improved expert retrieval results. Extensive experiments on two collections of the StackOverflow dataset prove the superiority of the proposed models compared to the state-of-the-art models. The results of the experiments show that the ASTM-2 model can outperform all baselines as well as the ASTM-1 model. The reason for this

superiority is the use of different context vectors in the ASTM-2 model. For future work, we intend to expand the proposed method in order to overcome its limitations. We also plan to try a new scoring approach for documents.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions which improved the paper.

References

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

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022.

Cifariello, P., Ferragina, P., & Ponza, M. (2019). Wiser: A semantic approach for expert finding in academia based on entity linking. *Information Systems*, *82*, 1–16.

Dehghan, M., & Abin, A. A. (2019). Translations diversification for expert finding: A novel clustering-based approach. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *13*(3), 1–20.

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

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

Fischer, T., Remus, S., & Biemann, C. (2019). Lt expertfinder: An evaluation framework for expert finding methods. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics (Demonstrations)* (pp. 98–104).

Fu, M., Zhu, M., Su, Y., Zhu, Q., & Li, M. (2016). Modeling temporal behavior to identify potential experts in question answering communities. In *Proceedings of international conference on cooperative design, visualization and engineering* (pp. 51–58). Springer.

Gharebagh, S. S., Rostami, P., & Neshati, M. (2018). T-shaped mining: A novel approach to talent finding for agile software teams. In *Proceedings of the 40th European conference on information retrieval* (pp. 411–423). Springer.

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

He, T., Guo, C., Chu, Y., Yang, Y., & Wang, Y. (2020). Dynamic user modeling for expert recommendation in community question answering. *Journal of Intelligent & Fuzzy Systems*, 1–12, Preprint.

Karimzadehgan, M., & Zhai, C. (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* (pp. 323–330).

Kundu, D., Pal, R. K., & Mandal, D. P. (2021). Topic sensitive hybrid expertise retrieval system in community question answering services. *Knowledge-Based Systems*, *211*, Article 106535.

Li, G., Dong, M., Yang, F., Zeng, J., Yuan, J., Jin, C., Hung, N. Q. V., Cong, P. T., & Zheng, B. (2020). Misinformation-oriented expert finding in social networks. *World Wide Web*, *23*(2), 693–714.

Liang, S. (2019). Unsupervised semantic generative adversarial networks for expert retrieval. In *Proceedings of the 28th world wide web conference* (pp. 1039–1050).

Liu, C., Hao, Y., Shan, W., & Dai, Z. (2020). Identifying experts in community question answering website based on graph convolutional neural network. *IEEE Access*, *8*, 137799–137811.

Ma, Y., Yuan, Y., Wang, G., Wang, Y., Ma, D., & Cui, P. (2019). Local experts finding across multiple social networks. In *Proceedings of international conference on database systems for advanced applications* (pp. 536–554). Springer.

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

Neshati, M., Beigy, H., & Hiemstra, D. (2014). Expert group formation using facility location analysis. *Information Processing & Management*, *50*(2), 361–383.

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

Neshati, M., Hashemi, S. H., & Beigy, H. (2014). Expertise finding in bibliographic network: Topic dominance learning approach. *IEEE Transactions on Cybernetics*, *44*(12), 2646–2657.

Neshati, M., Hiemstra, D., Asgari, E., & Beigy, H. (2014). Integration of scientific and social networks. *World Wide Web*, *17*(5), 1051–1079.

Nobari, A. D., Neshati, M., & Gharebagh, S. S. (2020). Quality-aware skill translation models for expert finding on StackOverflow. *Information Systems*, *87*, Article 101413.

Onan, A. (2019). Topic-enriched word embeddings for sarcasm identification. In *Proceedings of 8th computer science on-line conference* (pp. 293–304).

Onan, A. (2020). Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. *Concurrency Computations: Practice and Experience*, Article e5909.

Onan, A. (2021). Sentiment analysis on massive open online course evaluations: a text mining and deep learning approach. *Computer Applications in Engineering Education*, *29*(3), 572–589.

Onan, A., & Toçoğlu, M. A. (2021). A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification. *IEEE Access*, *9*, 7701–7722.

Ramanath, R., Inan, H., Polatkan, G., Hu, B., Guo, Q., Ozcaglar, C., Wu, X., Kenthapadi, K., & Geyik, S. C. (2018). Towards deep and representation learning for talent search at linkedin. In *Proceedings of the 27th ACM international conference on information and knowledge management* (pp. 2253–2261).

Saoud, Z., & Fontaine, C. (2018). Expert finding in citizen science platform for biodiversity monitoring via weighted PageRank algorithm. In *Proceedings of international symposium on intelligent data analysis* (pp. 278–289). Springer.

Shen, X., Jia, A. L., Shen, S., & Dou, Y. (2020). Helping the ineloquent farmers: Finding experts for questions with limited text in agricultural Q&A communities. *IEEE Access*, *8*, 62238–62247.

StackOverflow (2020). About - stack overflow. Online https://stackoverflow.com/company. Accessed 27 July 2020.

Tang, W., Lu, T., Gu, H., Zhang, P., & Gu, N. (2020). Domain problem-solving expert identification in community question answering. *Expert Systems*, Article e12582.

Tang, W., Lu, T., Li, D., Gu, H., & Gu, N. (2020). Hierarchical attentional factorization machines for expert recommendation in community question answering. *IEEE Access*, *8*, 35331–35343.

van Dijk, D., Tsagkias, M., & de Rijke, M. (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* (pp. 995–998).

Van Gysel, C., de Rijke, M., & Worring, M. (2016). Unsupervised, efficient and semantic expertise retrieval. In *Proceedings of the 25th international conference on world wide web* (pp. 1069–1079).

Wang, J., Sun, J., Lin, H., Dong, H., & Zhang, S. (2017). Convolutional neural networks for expert recommendation in community question answering. *Science China. Information Sciences*, *60*(11), Article 110102.

Xu, J., & He, R. (2018). Expert recommendation for trouble ticket routing. *Data & Knowledge Engineering*, *116*, 205–218.

Yang, C., Liu, T., Yi, W., Chen, X., & Niu, B. (2020). Identifying expertise through semantic modeling: A modified BBPSO algorithm for the reviewer assignment problem. *Applied Soft Computing*, *94*, Article 106483.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies* (pp. 1480–1489).

Yi, X., & Allan, J. (2009). A comparative study of utilizing topic models for information retrieval. In *Proceedings of the 31th European conference on information retrieval* (pp. 29–41). Springer.

Zhang, X., Cheng, W., Zong, B., Chen, Y., Xu, J., Li, D., & Chen, H. (2020). Temporal context-aware representation learning for question routing. In *Proceedings of the 13th international conference on web search and data mining* (pp. 753–761).

Zhao, Z., Yang, Q., Cai, D., He, X., & Zhuang, Y. (2016). Expert finding for community-based question answering via ranking metric network learning. In *Proceedings of the 25th international joint conference on artificial intelligence, Vol. 16* (pp. 3000–3006).