Engineering Applications of Artificial Intelligence 82 (2019) 126–147

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

The state-of-the-art in expert recommendation systems ☆

N. Nikzad–Khasmakhi, M.A. Balafar *, M. Reza Feizi–Derakhshi

Department of Computer Engineering, University of Tabriz, Tabriz, Iran

ARTICLE INFO

Keywords:
Expert finding system
Recommendation system
Information retrieval
Expertise

ABSTRACT

The recent rapid growth of the Internet content has led to building recommendation systems that guide users to their needs through an information retrieving process. An expert recommendation system is an emerging area that attempts to detect the most knowledgeable people in some specific topics. This detection is based on both the extracted information from peoples' activities and the content of the documents concerned with them. Moreover, an expert recommendation system takes a user topic or query and then provides a list of people sorted by the degree of their relevant expertise with the given topic or query. These systems can be modeled by information retrieval approaches, along with search engines or a combination of natural language processing systems. The following study provides a critical overview of existing expert recommendation systems and their advantages and disadvantages, considering as well different techniques employed by them.

1. Introduction

The rapid growth of the World Wide Web and the number of web pages has resulted in a tremendous increase in the amount of digital information/data and multimedia content. Hence, it has become more difficult for users to search their demands for the most related and newest information on the Internet (Su et al., 2013; Lousame and Sánchez, 2009). On the other hand, Information Retrieval (IR) provides a suitable and useful framework to find information, satisfying user demands. IR is the dominant form of information access methods that helps users to have access to those information related to user's queries. The query can be either a simple question such as "artificial intelligence related books", or a complex one such as "who is top-ranked in the music recommendation system" (Manning et al., 2008; Bobadilla et al., 2013).

IR approaches have been applied in miscellaneous applications on the Internet and social networks. As an example, we can denote recommendation systems that utilize IR approaches to obtain suitable knowledge by processing huge datasets in various formats. In recent years, recommendation systems have gained significant attention. Studies demonstrate their effectiveness in coping with information filtering and recommendation tasks. Recommendation systems have facilitated their usecases in manifold areas. The most noticeable studies are focused on music, television, books and e-commerce (Bobadilla et al., 2013; Su et al., 2013; Sun et al., 2015; Tang et al., 2013; Ge et al., 2015; Zhao et al., 2016b; Zheng and Li, 2011; Kim, 2013; Omran and Khorshid, 2014; Chandak et al., 2015; Al-Nazer et al., 2014; Tsuji et al., 2014; Ai et al., 2015; Ramezani and Yaghmaee, 2016; Chang et al., 2013). Approaches used in recommendation systems can be classified into three categories: collaborative recommendation system, content-based filtering, and hybrid recommendation. Collaborative recommendation systems filter and offer items based on measuring similarity. Based on the similarity, collaborative recommendation systems can be divided into two categories: user–user and item–item collaborative filtering. While user–user collaborative filtering technique makes recommendations based on contributions from other users in the same community; Item–item collaborative filtering focuses on finding similar items and not similar users (Linden et al., 2003). Some efforts have been made to improve the efficiency of collaborative recommendation systems. Including that, Pujahari and Padmanabhan (2015) proposed a new approach, called as Group Recommendation System, which combines user–user and item–item collaborative techniques. Content-based recommendation systems filter items based on the user's previous rating. In order to modify the recommendation accuracy of content-based recommendation systems, Ferdous and Ali (2017) and Boratto et al. (2017) proposed content-based filtering, based on the latent semantic analysis. They consider semantics behind an item description. Hybrid recommendation systems are the integration of two or more categories of recommendation strategies (Zhang et al., 2017). For reliable prediction in these three categories of recommendation systems, confidence is an important aspect that should be considered. Kagita et al. (2017) proposed conformal recommendation system, *CRS*, where conformal prediction framework guarantees the reliability of recommendations.

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.engappai.2019.03.020.

* Corresponding author.
E-mail addresses: n.nikzad@tabrizu.ac.ir (N. Nikzad–Khasmakhi), balafarila@tabrizu.ac.ir (M.A. Balafar), mfeizi@tabrizu.ac.ir (M. Reza Feizi–Derakhshi).

https://doi.org/10.1016/j.engappai.2019.03.020
Received 21 August 2018; Received in revised form 21 February 2019; Accepted 19 March 2019
Available online 8 April 2019

CRS assigns confidence values to the recommended items and finds a set of recommendations with specific confidence level ϵ.

With the advent of deep learning, the past few years have witnessed the tremendous success of the recommendation system in many online websites and mobile applications. For instance, a *Recurrent Neural Network* (RNN) based news recommendation system for *Yahoo News* or using the deep neural networks for *YouTube* video recommendation system are significant revolutions of this field in an industry with the help of deep learning (Zhang et al., 2017). In spite of the success of recommendation systems for estimating users' preferences on items, there are an array of challenges to improve the user satisfaction rate by offering the best items to one. One of these challenges is gathering information. Information has been considered as an important factor for the function and quality of the recommendation systems. There are various types of information and in order to recommend effectively, it is necessary to know that what information and which types of it, is appropriate for the system (Said et al., 2012; Isinkaye et al., 2015; Georgiou and Tsapatsoulis, 2010; Cao, 2016; Véras et al., 2015; Lu et al., 2015). Preference is counted as another challenge. The users' preferences are dynamic over time and may change depending on their current situations and purposes. These changes can significantly impact the performance of the recommendation systems for making suggestions accurately (Isinkaye et al., 2015; Georgiou and Tsapatsoulis, 2010; Cao, 2016; Véras et al., 2015; Lu et al., 2015). For example, a recommendation system may recommend a music convenient to one's current situation, based on being at a party or exercising, listening to music on TV and radio, using $MP3$ or listening from online music service providers. Describing these, the user's preference may be different (Véras et al., 2015). The cold start problem concerns the issue that the system does not have sufficient information about new emerging users or items to be used in recommendation process (Isinkaye et al., 2015; Georgiou and Tsapatsoulis, 2010; Cao, 2016; Véras et al., 2015). Scalability describes recommendation systems' capability to cope with and to perform suitably, in case of increasing the number of users or varying items. When faced with the growth of systems and large demands, a recommendation system should be able to maintain its level of performance (Isinkaye et al., 2015; Georgiou and Tsapatsoulis, 2010; Cao, 2016; Véras et al., 2015; Lu et al., 2015). The sparsity problem is related to the insufficient information about each item or user. This problem occurs because there are a limited number of items that are rated by users (Guo et al., 2017; Guo, 2012). In effect, sparsity is a major issue limiting the quality of recommendations. Of course, the similarity between two users is zero in collaborative recommendation system (Chen et al., 2011). Another drawback of the current recommendation systems is shilling attacks. Shilling profiles are injected into a system by an adversary. Since recommendation systems take the users' ratings into account, malicious vote ratings can cause serious damage. In other words, in order to affect recommendations, there may be users, the so-called attackers, who create false profiles and enter their votes in a biased manner. These kinds of damages are called shilling attacks. If attackers succeed, a users' trust in the recommendation system will be decreased. There are a lot of efforts done by researches, in order to overcome some of the mentioned challenges and improve the performance of recommendation systems such as studies in Said et al. (2012), Isinkaye et al. (2015), Georgiou and Tsapatsoulis (2010), Cao (2016), Véras et al. (2015) and Lu et al. (2015). Paper Zhang et al. (2018), by way of illustration, coped with data sparsity using combined group correlations and customer preferences. To construct a customer group, the similarity of the customers' preferences is the most important factor. However, the study investigated to balance the satisfaction of groups and individuals.

Recommendation systems also have had an enormous influence on knowledge management. A knowledge management system plays a significant role in making accessible the knowledge contained in documents (Yimam-Seid and Kobsa, 2003). It has also considerable functionality in specifying experts who have the most relevant knowledge about a particular topic (Zhen et al., 2012). Therefore, finding the appropriate experts in knowledge management system is a challenging issue. However, an expert recommendation system is a solution for dealing with this issue. An expert recommendation system takes the users' query firstly, next it gathers the past reputation of experts, then it classifies expertise into a subject classification schema, and finally provides a ranked list of experts that their expertise matches most closely to the user's query (Balog et al., 2012). In this way, an expert recommendation system can reduce costs for finding and selecting the best relevant experts in knowledge management systems (Zhang et al., 2007).

An expert recommendation system is a branch of general recommendation systems, hence it obviously has similar phases compared to general recommendation systems. The source of information is one of the most significant differences between expert and general recommendation systems. In contrast to the expert recommendation with no specific dataset, required dataset in music or video recommendation systems can be collected from websites such as *last.fm* and *YouTube*, respectively. Given the fact that people are often members of different social networks, it is more difficult to gather information about their knowledge and activity. Thus, the required information is collected by crawling various sites. Furthermore, other recommendation systems can also be used in expert ones; for instance, an expert recommendation system can use a follower recommendation system for *Twitter* to authorize experts. Moreover, an article recommendation system is able to find valuable articles related to the query and then it uncovers the associated experts with each article.

Another noticeable point is that expert recommendation system is a vital part both in academia and industry. One of the examples is *LinkedIn* where employers can search for potential candidates and job seekers can review the profile of hiring managers. Another example is social question answering networks such as *Quora*, *Stack Overflow*, *Stack Exchange*, *Yahoo! Answers*. These networks resort to expert recommendation systems for recommending users that have the most expertise in the question for answering. In academia area, *Expert Lookup* is an online tool that recommends experts who are really thought leaders in their fields (Zhao and Wu, 2016a).

Different manual and automated approaches are proposed for expert recommendation systems. Manual expert recommendation systems utilize expertise datasets which are updated by administrators or experts. Although these systems are very rapid in response and easy in terms of implementation, they suffer from issues such as costs of initialization, loading, maintenance and updating the datasets. In addition, the person who updates the information, may exaggerate the expertise. By comparison, automated expert recommendation systems extract expert information from updated sources by *IR* methods. These systems associate expertise with documents, web pages and CVs (Alarfaj et al., 2012). Notwithstanding the fact, these automated systems provide correct and updated information, building structures for these systems are complex and time-consuming (Wang et al., 2013; Gubanov et al., 2014).

The expert recommendation systems are satisfying both expertise-oriented and topic-oriented searching models. Different names have been used to refer these models such as "Candidate Model" and "Document Model" or "Candidate Generation Models" and "Topic Generation Models". In the expertise-oriented searching systems, the key goal is to find the fields with the highest similarities to the expertise of a specific expert. Expressed in a different way, the expertise-oriented models create a representation of experts and then rank them based on the query (Alarfaj et al., 2012). On the other hand, topic-oriented searching systems try to find an expert, in a particular topic, who is well-versed (Lin et al., 2017; Zhao et al., 2015). That is to say, these approaches find documents that are similar to the query; afterward, they detect the experts in these documents (Alarfaj et al., 2012).

What are the differences between this survey and previous ones? The success of expert recommendation system requires a review for successive researchers to better understand the weaknesses and strengths of such systems. There are a number of studies in the field of expert recommendation system. To the best of our knowledge, there are very few reviews which shape this area, summarize current efforts and describe the open problems present in this scope. Lin et al. (2017) presented a survey of existing expert recommendation systems along with the key issues in the field of expert finding including resource selection, expertise retrieval, and retrieval model extending. The authors summarized the state-of-the-art methods for each issue and analyzed the limitations of the existing methods. Moreover, authors in survey (Al-Taie et al., 2018) reviewed the current researches for the expert finding task in online communities and corporations. Authors classified expert finding systems based on two criteria: domain and methods. Based on domain, studies were categorized into the organization and online environments groups. Otherwise, expert finding methods were divided into two classes: graph techniques and machine-learning techniques. This paper did not point out the open issues and promising future research directions. Wang et al. (2018) presented an overview of the research efforts for the expert recommendation in Question Answering Communities(QAC). Although, authors summarized and compared the existing methods based on aspects such as datasets, input and output, and evaluation metric, the survey did not completely cover deep learning approaches employed in QACs. Our survey provides a comprehensive summary of the state-of-the-art expert recommendation systems. However, we focus only on automated and topic-oriented expert recommendation systems. We conduct a review on recommendation models and propose a new classification scheme to organize the current researches. We also provide an overview of the state-of-the-art studies and summarize their advantages and disadvantages. Moreover, our study discuses the challenges and future research directions.

Contributions of this survey. The goal of this survey is to thoroughly review the literature on expert recommendation systems. Indeed, researchers and educators who are interested in expert recommendation system, can use this survey. This survey firstly begins with defining some key concepts and introducing the basic elements of an expert recommendation system. Secondly, the paper proposes a procedure including typical phases of the expert recommendation system. What is more, it overviews the current applications of expert recommendation systems. On the other hand, it clusters the existing expert recommendation methods according to their characteristics. Moreover, this study reviews and analyzes researches based on the datasets and approaches that use in their works. Also the advantages and disadvantages of approaches are discussed. Further, this paper provides a list of metrics to evaluate expert recommendation systems and shows the results of these metrics that are obtained from previous studies. At the end all the things considered, the survey enumerates the existing challenges and has an outlook on promising directions for future researches to solve these issues.

The remaining sections of the paper are organized as follows: we describe an introduction of expert recommendation systems and define some key concepts in Section 2. Also in this section, the paper proposes a procedure that shows the phases involving in the expert recommendation systems. Section 3 overviews current expert recommendation system applications. Section 4 presents our classification framework. In Section 5, the ground truth and evaluation metrics are described. Additionally, Section 6 illustrates the state-of-the-art experimental results. In Section 7, the survey discusses the challenges and prominent open research issues. At the end, Section 8 concludes whole paper.

2. Expert recommendation systems

Among the various information retrieval application domains, "*Expertise Retrieval*" is an emerging one that leads to an expert recommendation system. The expert recommendation system is also called an expert finding system. An expert recommendation system attempts to detect the most knowledgeable people in some specific topics. This detection is based on both the information extracted from peoples' activities and the content of the documents concerned with them. Moreover, an expert recommendation system takes a user topic or query and then provides a list of people sorted by the degree of their relevant expertise with the given topic.

Due to the fact that the concept of "expert" is not clear, it is hard to identify the expertise areas of an expert (Moreira and Wichert, 2013). In addition, the growth of the Internet and information resources cause researches to have different opinions about the definition of expert. To tackle this issue, numeric scores, represented by y_i, are assigned to each of the experts that demonstrate their level of expertise.

Because there is no formal explanation about expert and expert recommendation system in the literature, it motivates us to represent comprehensive and scientific definitions of them. These definitions are used in the entire paper. Hence, in the following we explain the necessary definitions that who an expert is and what an expert recommendation system does in detail.

Definition 1. User u_i is called an expert if and only if his/her score is higher than the threshold θ, as described in Eq. (1):

$$u_i \text{ is an expert } \Leftrightarrow y_i \geq \theta \tag{1}$$

or

$$\forall y_i \in Y \Leftrightarrow y_i \geq \theta \tag{2}$$

Each user u_i is explained with a set of ordered pairs (x_i, y_i), where x_i is a feature vector of one's expertise in the Space X (expertise feature space). These features are extracted from the users' activity on social networks and textual information published by them. Similarly, y_i declares the score of each expertise for an expert. In Definition 1, Y is the ground truth ranked list of experts. The ground truth refers to the desired ranked list of experts that is provided by direct observations. The value of threshold θ is determined depending on the application scenarios.

Definition 2. expert recommendation problem is initiated as finding a ranked list of experts y'_i from list of features x_i based on training dataset $\{(x_i, y_i)\}_{i=1\rightarrow Datasize}$.

In Definition 2, $y'_i \in Y'$ is the score of user u_i that is predicted by the expert recommendation system. Y' indicates the predicted ranked list of experts which is the output of the system.

After providing the required definitions, we will study the general structure of the expert recommendation system. Fig. 1 indicates the basic elements of an expert recommendation system. All exhibited expert recommendation systems, have similar basic inputs; including a database and a query topic. The database input comprises the expertise feature of experts, X. Although, it can include users' shared textual content and social activities, but more input might be available depending on the application scenarios. For user u_i, $x_i \in X$ contains shared questions, answers, and documents by this user, and his/her connections in social networks. More details of these inputs are described in the content related to the information retrieval of experts. The query topic input is designed to enter search queries; namely the system will filter the experts based on keywords existing in the query topic.

The notable difference among expert recommendation systems are their learner element. As it is seen in Fig. 1, the functionality of the learner element is analyzing the inputs and finding the experts with the most similar expertise to the query topic. It means that, the learner maps the expert's expertise, x_i, to his/her corresponding score, y'_i. As indicated, a feedback, $ErrorFunction$, exits between the learner and output. $ErrorFunction$ feedback helps the learner in error correction(which is a minimization problem). The predicted output Y' is compared with the desired output Y. When predicted output Y' is

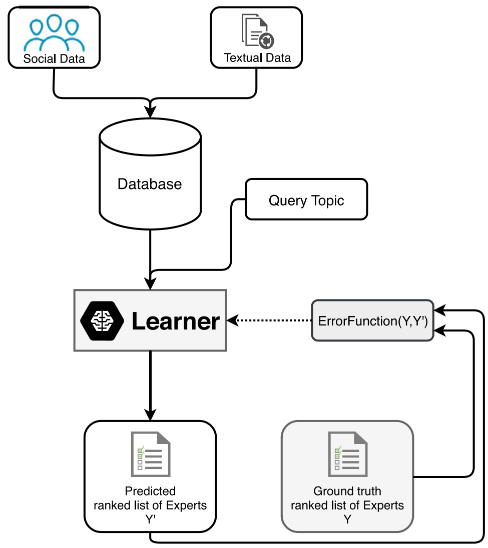

Fig. 1. Basic elements of an expert recommendation system.

equal to the ground truth ranked list Y, then $ErrorFunction$ is zero. Therefore, there is no need for more training or error correction. If Y' is not the same as Y, then continuous error correction process is repeated until $ErrorFunction$ value becomes zero or training algorithm stops at specific error threshold.

The learner approaches can be classified as Supervised and Unsupervised. Supervised learning tries to learn relationships and dependencies between the target prediction output and the input features. The supervised learner attempts to minimize $ErrorFunction$ as shown in Eqs. (3) and (4):

$$Error_{Y,Y'} = \Psi_{i=1}^{|Y|} ErrorFunction(y_i, y_i') \tag{3}$$

$$\lambda = argmin(Error_{Y,model(\lambda,X)}) \tag{4}$$

In Eq. (3), Ψ refers to a summation function like average or sum over all of Y and respective Y'. Eq. (4) is the actual learning process in which the learner tries to optimize the $model$ by finding optimal parameters noted as λ. These parameters are compared by evaluating $Error$.

On the contrary, unsupervised learning only receives input data and does not have any feedback from its output. That is to say, both predicted and desired ranked list of experts are provided in supervised learning that causes accurate and reliable outputs. Nevertheless, the results of the unsupervised task are moderately accurate and reliable because there is just a predicted ranked list of experts.

By looking at the elements of Fig. 1, their functionalities are investigated in the following. Hence, we propose a diagram, as illustrated in Fig. 2, that shows the typical stages of an expert recommendation system. We suggest this diagram after perusing the existing procedures in current expert recommendation systems. To the best of our knowledge, Fig. 2, covers all of the existing expert recommendation systems.

Information of experts goes through a series of stages before they emerge as ranked experts for recommendations. First, appropriate sources are selected that comprise experts' textual and social activities information. The next stages depend on the source type, i.e. being off-line or online. In the off-line sources, it is just necessary to download the dataset from the specified source. While, in online sources, the dataset should be collected and built from web pages by crawling. The output of the second stage can be divided into three categories: Document, Question and Answer, and Social graph dataset. In the next stage, query and datasets are given as inputs to the information retrieval system. After this, an *IR* technique retrieves the information based on the query from the datasets. Due to the nature of Internet information that is unstructured, text information requires preprocessing. Therefore, the next stage is information preprocessing. Then, the expert behavior pattern is extracted from the information. At this instant, experts are ranked based on their degree of similarity of the behavioral model with the query. The general stages in an expert recommendation system are as mentioned above. For better understanding, this survey divides these stages into two phases, namely information retrieval of the expert and predicting the expert's level of expertise. Left and right blocks in Fig. 2 demonstrate two major phases and their stages. According to this categorization, the major function of the first phase is filtering information and acquiring all relevant information of experts and building the input database in Fig. 1. The second phase tries to find the expert behavioral model through the information collected from the previous phase and to rank experts in a particular field. That is to say, the second phase does the learner process and provides the predicated list of experts. In the following subsections, details for each phase of an expert recommendation system are discussed.

The core of any decision support system is information gathered from different sources. Expert recommendation systems also follow this rule. Left block in Fig. 2 shows the first phase of an expert recommendation system. As shown, the retrieval of the expert's information consists of three stages that are information source selection, retrieval and preprocessing. At first, the appropriate sources are identified to gather the required information. In the second stage of this phase, suitable techniques are proposed to retrieve required information from the selected sources, based on the query. Stop word removal and stemming are two important tasks that done in the third stage. As follows, these stages are described.

One of the main parts of each recommendation system is the collection of information. If it were done in a regular and accurate manner, the analysis of data will be accomplished with great speed and accuracy. With the advancement of Internet, although users have choice to select from a large number of information sources, based on their preferences and requirement, searching and selecting the appropriate sources require more time and effort. Therefore, choosing sources which are most likely to contain relevant and reliable information for a query is a critical issue. Sources of information can be classified based on some characteristics (Johnson, 1986). Some of these characteristics are online, architecture, quantity and formal which are briefly described in the following section:

On-line: Sources in the expert recommendation systems can be defined as an off-line dataset or an online data stream. If the selected source is off-line, then there is no need for crawling stage and the dataset is downloaded from the source. On the other hand, for the case of online data stream, a *crawler* is implemented to gather information from selected sources and to create the datasets.

Architecture: Based on how the information is organized in the sources, the architecture of sources can be classified as a source with unstructured or structured information. The sources with structured data impose a particular structure or pattern over the information. Also, the structure of information is under the control of certain strategies. However, since the organization of information is firmly controlled, the cost of maintaining the structure of information is high. On the other hand, there is no structure globally imposed upon sources with unstructured data. Even though, unstructured information can be easily implemented, it is inefficient to find the desired information due to wasting too much of time (Balog et al., 2012; Wang et al., 2013; Johnson, 1986; Chen et al., 2013). In summary, unstructured sources have slow search and large access time while structured sources are more feasible in terms of search but infeasible in terms of data maintenance. The information retrieved from the Internet is unstructured and does not have an explicit and semantically obvious structure for modeling and rating (Steichen et al., 2012). Thus so many techniques are proposed to find patterns for unstructured information and interpret this information such as *natural language processing* and *data mining*.

Quantity: The source size is another aspect to characterize information sources. For instance, in the small local area networks such as small

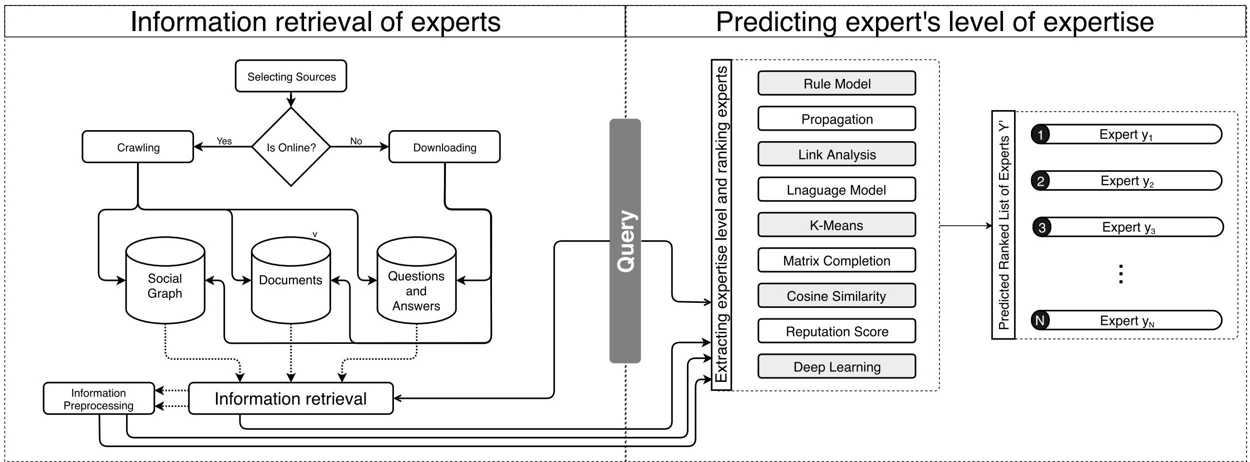

Fig. 2. The procedures of expert recommendation system.

academic research networks, the quantity of information is small and its quality is high. In these sources, information is limited into those inserted by administers or editors. Hence, information is not updated and not relevant to the last changes. On the other side, the wide area networks provide a rich information space in which sources can be discovered and retrieved. These environments prepare a framework in which expert can connect to other experts with similar professional interests (Wang et al., 2013; Zhao et al., 2015; Chen et al., 2013; Lamond et al., 1996). With respect to the nature of big-data in these networks, analysis, indexing and retrieval can be solved by the use of tools such as apache *Luecene* and *Hadoop* that is free and open-source information retrieval software. The wide area networks include online Knowledge Communities (*KC*s) or *QAC* systems.

Formal: Sources can also be categorized based on their styles as *formal* or *informal*. Reports, archives, journals, conference proceedings, books, bibliographies, and datasets are examples of formal information (Johnson, 1986). Unlike formal information, informal information is more personal and difficult to characterize. Furthermore, informal information is used more in *QAC*s or *KC*s.

The above characteristics of sources make it difficult to recognize and select appropriate sources and consequently the valuable information cannot be retrieved. An expert finding system needs appropriate data in terms of quality and quantity.

As shown in Fig. 2, after collecting the desired information having the above characteristics, it will be divided into three categories: Document dataset, Question and Answer dataset, and Social graph dataset. Document dataset consists of information about expert's shared papers such as abstract, keywords, the number of citations, and etc. Question and Answer dataset includes the content of the question asked or the answer replied by the expert, the question or answer topics, the best answers, and etc. These two datasets represent textual data in Fig. 1. The document dataset is used to find academic experts. However, Question and Answer dataset is utilize to find experts in *QAC* systems. Social graph dataset encompasses information about the experts' relationships in social networks. These relationships can reflect the co-authorship in the academic environment or the connections between questionnaires and answers in *QAC*s and etc. These datasets provide the feature vector of expert's expertise, x_i.

In last years, the increasing content of the web and the difficulty of access to the content have motivated *IR* techniques. Accordingly, *IR* approaches help users to retrieve information that is relevant to their interests and preferences. One essential task in the recommendation system is the ability to find appropriate information. *IR* technique attempt to extract the necessary information about experts from the datasets which are created in the previous stage. Moreover, retrieval techniques are selected based on the requirements of the recommendation system.

In the expert recommendation system, as shown in Fig. 2, the output of the first stage is divided into three different datasets: Document, Question and Answer, and Social graph. Two first datasets are types of text and Social graph represents interactions between experts. As a result, *IR* techniques are divided into two categories based on these datasets calling Content-based Information Retrieval (*CBIR*) and Social Graph-based Information Retrieval (*SGBIR*). *CBIR* retrieves the content of user's shared items based on the similarity of the published content to the query. It means that *CBIR* compares the characteristics of the query and the text content. Shared items include either content of documents, the content of the questions asked or the answers replied by the user. On the other hand, *SGBIR* extracts the communications between experts in social networks. In recent years, social networks are increasingly being used as an information source where their information is often unstructured and informal. These information sources have provided simple facilities for users to generate and share a variety of information. In the expert recommendation system, one part of the expert's information is obtained from expert's relationships with other experts who like the same topics and have similar preferences (Bonnin et al., 2008). As mentioned, one output of the previous stage is Social graph dataset that is derived from the connections of experts with each other in social networks. A major question surrounding the use of social networks as an information source is how to retrieve information from them. A graph structure has become more and more important technique to represent and efficiently and effectively retrieve the required information from social networks with complicated structures. In a graph $G = (V, E)$, nodes V are a set of experts and edges E represent the relationships between experts in the social networks. Type of relationships can be question-answering, coauthor, colleague, friend and so on.

Information preprocessing, as final stage of this phase, is an important task in Text mining and *NLP* (Kannan and Gurusamy, 2014; Gupta et al., 2009). Stop word removal and Stemming are two common tasks of information preprocessing that are mostly used in finding expert systems. The stop word removal prunes of non-essential words (*e.g.* the, a, an) that occur commonly across all the documents in the corpus (Uysal and Gunal, 2014). Coming to stemming task, input text will be segmented into greater meaningful segments based on common punctuations in language (Jivani, 2011). For instance, a user may search for the term "organization" and stemming task may return query results for any word that contains the root form of the word (e.g. organize, organizes, organizing) (Manning et al., 2008).

Providing good and useful recommendation depends on the system capabilities for learning and modeling user preference. As the user behavior may change in a period of time, and it is important for recommendation systems to understand the user's updated behavior which it helps to recommend appropriate items. Right block in Fig. 2, the second phase of an expert recommendation system, extracts an expert's behavior patterns and doing the tasks of the learner element. In this phase, the learner is responsible to analyze the feature vector of experts' expertise and extract their scores. Hence, employing an efficient and accurate learner technique is very important. For finding experts in *QAC*, an effective approach for extracting the expert's patterns of expertise should contain three different scores: *Text similarity* (S_{TS}), *Reputation* (S_R), and *Authority* (S_A) score. These scores are combined to derive an expert's score. Finally, the output is a ranked expert list that is created based on the experts' score.

Moreover, it is the text similarity score that compares the characteristics of the content contained in the published items associated with a user and the query. This score represents the level of expert's knowledge in the field of the topic query. The expert's reputation score is obtained from his historical activity in social networks. Eq. (5) shows the reputation score that is a function of answers and the best answers given by an expert. Thus an expert who provides more answers and the best answers, has high reputation score and consequently shares more knowledge with others in the communities.

$$S_R = g(Number\ of\ Answers, Numbers\ of\ Best\ Answers) \tag{5}$$

Coming to Authority score, it is the expert's measure of influence and popularity in social networks; i.e. the more influential expert is, the one whose authority is the highest. This score is determined by some factors such as: how often the expert does posts on the social networks, who the expert's friends are and how much his friends are heavy influencers. The learner maps the database input into these three scores and applies a combination strategy, as Eq. (6), that combines these scores into a single score y_i' and lists experts based on y_i'. The scores can be combined in different ways such as a weighted sum or multiplication (Wang et al., 2013).

$$y_i' = f(S_{TS}, S_R, S_A) \tag{6}$$

Besides, the main objective of an expert recommendation system is to determine the expert's knowledge areas and find the appropriate experts, the output can be different based on the application scenario. As an illustration in *QAC*s, while some studies attempt to provide high quality answers from the best answerers, others have the intention of achieving a ranked list of the best answerers.

The benefits of the expert recommendation system are well-recognized both in industry and academia. Correspondingly, for the success of an organization, it is essential to identify experts. Utilizing an expert finding system, an organization can speed up the process of conducting research with the rapid formation of operational teams (Singh et al., 2013). Experts with overall knowledge monitor the research priorities of an organization and help the organization to achieve its rightful places. Furthermore, manufacturers need to have expert human resources to properly utilize industrial approaches and to take ultimate advantage of the technologies. Editors of journals try to choose the right reviewers who are knowledgeable experts for submitted research proposals and manuscripts. In like manner, identifying someone as a keynote speaker for a conference, an expert recommendation system provides an immediate and professional response (Balog et al., 2012). Moreover, expert finding systems are an essential part of *QAC*s. They act to increase the quality of these communities by choosing the experts and suitable users for answering the questions (Elalfy et al., 2018). Nowadays *QAC*s are being extensively used, to share and exchange knowledge. Due to the nature and characteristics of the recommendation systems, they are really time saving and helpful to choose the best.

3. Expert recommendation system applications

Raise of intelligent systems and recommendation techniques plays an important role in the developments of expert recommendation system applications. Consequently, there is various applications based on expert recommendation techniques. In this section, the developments and applications of expert recommendation systems are reviewed. Moreover,a classification of applications including general purpose and specific purpose will be described.

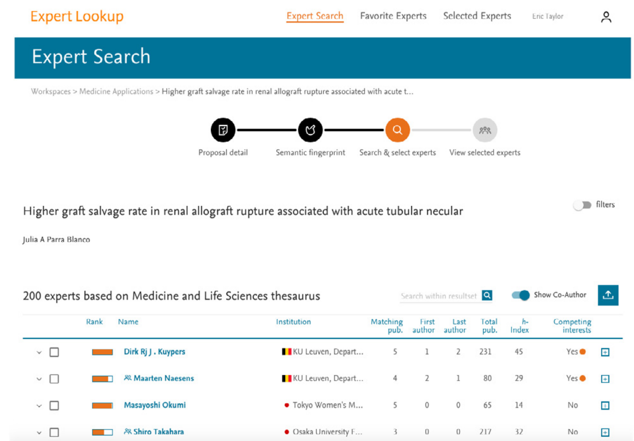

Fig. 3. Expert Lookup (Zhao and Wu, 2016a).

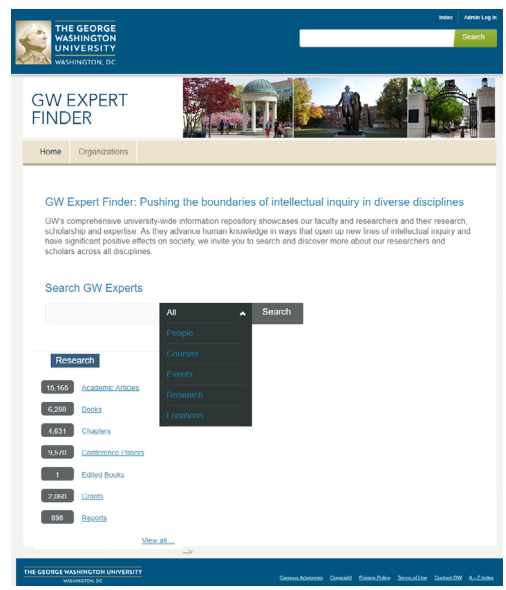

Fig. 4. GW Expert Finder (The George Washington University, 2018).

- General purpose applications
  A general-purpose expert recommendation system application is one that, given an appropriate application, performs well on different domains. An overview of some typical general purpose applications is presented in the following.
  Web services are an important and popular general purpose application to find experts over the web and to generate recommendations based on their expertise. An expert search engine is a web application that is designed to search for experts. Expert Lookup is an expert search engine that is powered by *Elsevier's Fingerprint Engine*. *Expert Lookup* helps agencies, research institutions and corporations to identify scientific experts. It uses the state-of-the-art Natural Language Processing (*NLP*) techniques to extract key concepts from unstructured text to ensure that the recommended experts are relevant and truly thought leaders in their fields (Zhao and Wu, 2016a). Fig. 3 represents the output of *Expert Lookup* that generates a list of experts with indexes of expertise. Fig. 4 shows

GW Expert Finder search engine. It is *George Washington*'s comprehensive university-wide information repository that showcases faculty and researchers and their research, scholarship and expertise (The George Washington University, 2018). Expertise Finder is another expert search engine built for the mobile era to help experts to be seen in a positive light (Rougas and Ashrafizadeh, 2018).

Additionally, e-commerce websites, as another type of web application, provide an environment in which users, called consumers, can comment or review products online. Valuable comments can motivate other consumers' purchase intention (Wang and Wang, 2018). With the passage of time, a huge amount of online reviews generated by diverse users are accumulated and this can introduce problems raised from review reliability and user reputation (Faisal et al., 2018). Therefore, the reviewers in e-commerce websites can be considered as experts and an expert recommendation system can be used to guarantee the reliability and credibility of reviewers in these applications. The goal of expert finding in e-commerce applications is to find consumers who can provide high-quality and useful comments or reviews to a new product or service (Faisal et al., 2018). Paper Wei et al. (2015) finds experts in online forums of Epinions.com, a product review website. The authors' proposed method relies on the opinion ratings from the members in the forums. Paper Wang et al. (2017b) proposed a new approach that investigated the effects of the perceived value of consumer characteristics on the influence of reviews in online recommendation. The authors extracted personalized perceived value of consumers using a clustering method. Another interesting point in e-commerce websites is that the reviewers may give their opinions about a statement with diversity in evaluation; for example a reviewer's opinion can indicate that the related statement is 50% True, 10% False with 20% uncertainly. Hence, it is difficult to evaluate this qualitative information using a numerical method. One of the solutions for this problem is fuzzy-based approaches that describe the fuzzy preference information of reviewers about objects (Wang and Wang, 2018; Liang et al., 2017).

Question-answering services provide another way to establish expertise in *QAC*s. These services such as *ResearchGate*, *Zhihu*, *Yahoo! Answers* and etc. make use of expert recommendation systems for suggesting users that have the most expertise considering the topics of answers and questions. To give an instance, in *ResearchGate*, when an user searches for experts of a special topic, *ResearchGate* explores objects including the latest articles, projects, questions and answers on this topic, and then calculates experts' score. At the end, it lists the top experts based on their scores.

Although there are not any practice of applying expert recommendation system in other engineering fields such as civil engineering, aerospace engineering and etc., but it has the capability to be extended to these fields. For example in civil engineering, finding knowledgeable and experienced pavement engineers who can accurately identify the type and general causes of deterioration exist in the pavement leads to a pavement evaluation and rehabilitation expert system. This system elicits knowledge from experts and makes automatically decision for the process of observation and pavement rating (Akram et al., 2014). In another case, expert recommendation system can be used to find consultants. These consultants have the ability to consult regarding issues involving the bridges, sewers, flood control, and earthwork. Also, these experts can present expert comments concerning site engineering, water supply, sewer service, and electric and communications supply. Further, an expert recommendation system can help to find experts who can provide remarkable opinions regarding the practices of electrical engineers and the electrical safety code. These experts present reports about electromagnetic compatibility, voltage regulating devices, switching equipment and electronic engineering. We also need an expert recommendation system to search experts who have an opinion about applied mechanics, structural analysis, and mechanical failures. These experts enable to design or redesign mechanical and thermal devices. The output of the expert recommendation system in aerospace engineering is a list of aerospace engineers who have real world experience including ensuring the quality standards and safety of aircraft and aerospace technology, designing, developing, and fabricating of such products. Moreover, an advisory system is a tool that supports the making of decisions. Principally it is a version of an expert system specialized in the task of advising (Gajzler, 2017). Advisory system provides expertise to support decision making (Beemer and Gregg, 2008). As an example of usage of advisory system is in agricultural engineering. Research Kassim and Abdullah (2012) proposed software architecture for advisory systems. Farmers as a client ask an advice from the experts to help them in making decision process in their cultivating. Advisory system take their knowledge from the observations that are defined by human experts who work in this domain. That means expert and advisory system can be used successfully for design, diagnosis, and monitoring if human experts are ground truth of these systems. So, an expert recommendation system can introduce knowledgeable people as ground truth. On the other hand, the expert recommendation system can offer a set of consultants to form a team for a particular project, in all engineering fields. So, it is necessary that these consultants should be at the same level to support and cover each other and also collectively work with each other to perform the tasks in the projects. For example, JurisPro is a search engine for experts in a specific field and location by selecting a state. This search engine can find experts in all engineering fields. Fig. 5 shows a list of aerospace engineering experts in *Washington* state that can form a team of experts for performing a project.

Most of the existing expert recommendation system applications have been developed to determine experts in environments such as academic, organizations, social networks and QACs (Neshati et al., 2017). In addition, future expert prediction can be a useful application that predicts the direction of experts' expertise. It determines the future expertise of an expert based on one's current performances. It is able to detect one's who have potentially the facilities to become an expert in the future. Predicting future experts with full accuracy is a difficult task because people usually change their interests and expertise topics over the time (Neshati et al., 2017).

- Specific purpose applications

In contrast to general purpose application, specific purpose applications of expert recommendation systems provide services that are limited to particular topics or domains. One of these domains is health-care area. As a case in point, MedHelp is an online health community where patients ask their medical questions and experts from different hospitals and medical research institutions answer these questions. As another case, an expert recommendation system can process the information stored in Electronic Health Records (*EHR*) to recommend the best matching specialist to a patient (Lopez-Nores et al., 2011). *EHR* refers to the use of the Internet and other information and communication technologies to collect patients' health information. In this case, patients' preference like physical examinations are indicated as a query topic and a specialist is identified as an expert. The other application of expert recommendation systems in health is physician recommendation. Paper Sahney and Sharma (2018) proposed a physician recommendation system using underlying evidence-based ontology. The researchers considered the patient's medical conditions and pain description characteristics as patient preferences.

Among other things, expert recommendation systems can be exploited in software engineering. Experienced architects and developers play a crucial role to address stakeholders' concerns during the software development phases (Bhat et al., 2018). From this perspective, an expert recommendation system application can be helpful in quantifying architectural expertise of software architects and developers. Paper Bhat et al. (2018) proposed an expert recommendation system that identifies developers who have expertise related to particular shortcomings in software systems to engage them in software engineering projects. Additionally, agile software development, as a successful approach in software engineering projects, can manage human resources and use expert recommendation system to retrieve candidates in forming agile teams. In this viewpoint, paper Rostami and Neshati (2019) introduced agile team formation as an expert finding problem. In this paper, a set of skill-areas of an agile team is given as input and the ideal output is a team of candidates who are specialist in a specific topic and have general knowledge in other topics of the team. In the same manner, paper Bayati (2016) proposed a framework for finding expert software engineers who have expertise in information security.

The expert recommendation system can also be transferred into business domain. Job search engines are a type of the expert recommendation system in which the query is the employers' hot topics, experts are identified as job seekers, patterns containing data of skills and abilities are identified with user personalized resume. Some applications have been developed such as Indeed, CareerBuilder, LinkedIn, and Google Careers. *Indeed* is a specialized and highly popular job search engine. Currently, potential employers are strategically finding suitable job candidates by searching them out on *Indeed* (Gross, 2012).

CourseRank, developed by Stanford students, is an expertise-oriented application which recommends the right courses to each student. Inspired by this application, paper Engin et al. (2014) proposed an application that is a course advising system which suggests courses to students. This paper also advanced a scholarship recommendation system for undergraduate students based on their eligibility.

ExpertSeer is a framework for expert recommendation based on the contents of a digital library that is proposed by Chen et al. (2015b). The framework is built in order to recommend experts in computer science and chemistry areas, namely *CSSeer* and *ChemSeer*. *CSSeer* uses the *CiteSeerX* digital library to recommend experts in computer science. *ChemSeer* utilizes documents that available in the*Royal Society of Chemistry* to suggest experts in chemistry.

A few of the mentioned applications are utilized in real-world. Some implementation issues can arise during developing these applications. One of these issues is the shortage of appropriate data sources. As a case in point, supervised learning approaches need labeled data for learning and it is hard to find this data in real-world. Along with, the cost of assigning labels to the data is expensive and needs human efforts. The underlying challenge is to model a learning task that uses both labeled and unlabeled data. As another issue, the current applications use specific databases to gather experts information that are fixed and they just considered limited attributes of experts. For instance, *Expert Lookup* employs *scopus* database which is *Elsevier*'s abstract and citation database. From another point of view, reliability and speed are the problems posed by crawling web pages to retentive different kind of information about experts. Next issue is choosing to design either a supervised or unsupervised learner. The decision typically depends on factors related to the structure of data and the use case. In addition, the expert recommendation systems face with the challenges that mentioned in the introduction. Suppose the effect of shilling attacks on *MedHelp*. Injecting fake answers into this application causes inappropriate therapy that result in patient mortality. Another cases that should be considered in expert recommendation system applications is that the experts' preferences change depending on their current situations and purposes, thus their patterns of expertise are dynamic over the time. Alternatively, an expert recommendation system should modify itself according to these changes. To give an example, a general practitioner becomes a specialist in a type of diseases and simultaneously, he joins *MedHelp* and shares his knowledges with others. An effective expert recommendation application should notice these changes in order to rank accurately this expert (Cao, 2016; Véras et al., 2015; Lu et al., 2015).

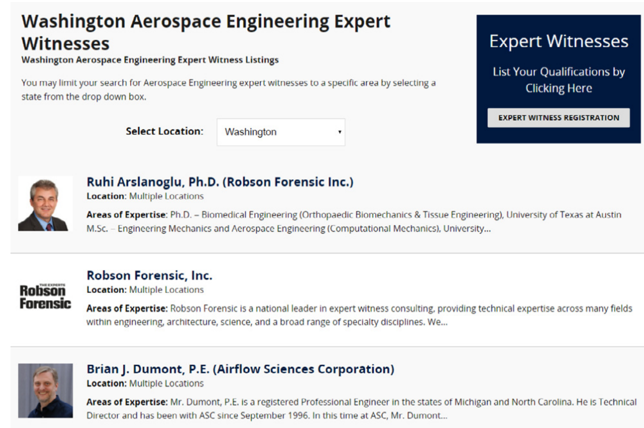

Fig. 5. Aerospace engineering experts in *Washington* state.

4. Critical review of the state-of-the-art in expert recommendation systems

In this section, we investigate and classify the state-of-the-art in expert recommendation systems. Firstly, we summarize the most used sources in expert recommendation system related researches. After that our classification framework is presented. We categorize existing publications, based on their different approaches that are used to extract experts' behavioral patterns: Rule based model, Propagation, Link analysis model, Language model, K-Means, Matrix completion, Vector Space Model, Reputation score and Deep learning approach. In the next part, the studies are represented that include these approaches. Finally, we discuss the advantages and disadvantages of each class.

The following sources are utilized in the most researches. *Yahoo! Answers*, *Stack Overflow* and *Quora* are attracted attention due to their applications to find experts in question-answering systems. Expert social activities can also be examined by the assist from *Twitter* dataset.

- Yahoo! Answers
  Yahoo! Answers is a knowledge market from Yahoo!. It is a question-and-answer website that questions are asked and answered by its community of users. It provides a suitable dataset for finding experts in *QAC*s.
- Stack Overflow
  Stack Overflow is another question-and-answer website for the expert recommendation in *QAC*s. In *Stack Overflow*, developers can learn and share their knowledge on a wide range of topics in computer programming.
- Quara
  Quora is a community-driven question-and-answer website that allows users to submit questions to be answered and answer questions asked by other users.
- Twitter
  Twitter is one of the online news and social networking services where users can post tweets. Users interact with each other

through tweets. *Twitter* contributes the required dataset to analyze an expert's social network activities.

- Enterprise track of TREC
  The Enterprise Track includes all the **.csiro.au* (public) websites as they appeared in March 2007. It considers the experiences of users in real organizations. The fourth year of the enterprise track, 2005–2008, has published with tasks: expert, e-mail known item, e-mail discussion and document search (Balog et al., 2008).
- Other datasets
  There are other datasets that are used for the study of expert recommendation methods. These datasets include collaboration network such as *DBLP* and *Google Scholar*, social networks such as Epinions.com that allows members to share their reviews about different products, *Microsoft Office Discussion Groups*, digital libraries such as *CiteSeerX* and *Wikipedia*.

Table 1 demonstrates the sources that used by different studies and describes their characteristics including online, architecture, quantity and formal. It is necessary to mention that datasets with the size larger than 1000 are considered as large datasets in this survey. Moreover, Table 2 includes the techniques used for retrieving information in each research. By examining the type and sources fields in Table 2, it can be concluded that the most current studies are focused on finding experts in *QA* systems.

As mentioned, the important part of an expert recommendation system is the learner element. This element tries to model the experts' behavior and calculate their related scores. A number of different approaches have been proposed for this goal. The following content obeys this scenario: first the state-of-the-art approaches are firstly classified into nine categories. Then, a brief explanation of these classes is presented. After introducing all categories, the previous studies are investigated that apply these approaches in their works. The last part has conducted a survey of the advantages and disadvantages of each category. and discuss the characteristics of each category.

In general, *rule-based* systems are associated with decision making in structured contexts. These systems represent knowledge in terms of a set of rules that determine what to do or what to conclude in different conditions (Grosan and Abraham, 2011). Rule-based systems can collect and process real-time data and discover patterns and relationships between them. A rule-based system is based on a cluster of facts, a set of "if-then" statements, and have some interpreters that control the set of actions that are executed. In addition, there are two kinds of rule systems: forward chaining and backward chaining systems. The former starts with the initial facts and looks back to make new conclusions. It determines which "if" rules to use. On the other hand, the second type starts with some goals and looks back to find actions in the "then" rules. It means that forward chaining systems are primarily data-driven and backward chaining systems are goal-driven systems (Alison, 1997).

A propagation approach is the distribution of all scores of experts that is managed with a certain data structure such as a graph or a linked list.

Link analysis is another approach that is employed by most studies to compute Authority score. Nowadays, hundreds of millions of people participate in social networks and hence the social networks have attracted a widespread attention as a rich source of information. Analyzing these networks can provide both information for evaluating users' activity patterns and their relationships in recommendation systems. Several analysis algorithms are proposed for this purpose such as link analysis. It is a technique that has the ability to analyze the relationship between two entities and find their patterns in social communications and social networking websites. Entities may have various types, including pages, organizations, people, and transactions. In a graph $G = (V, E)$, where V and E can be the set of entities and is the links between entities in the social networks, respectively. For example, in the link analysis of hyper-links, web-graph is represented as a directed graph that each web page acts as a node of the web-graph and each hyper-link on the web is a directed edge to its destination page (Alyguliev, 2007). To evaluate relationships (connections) between pages, link analysis has been used as an effective tool for search engines (Zhang et al., 2007; Ströele et al., 2013; Ng et al., 2001). Search engines measure the similarity between web pages with a topic query and then rank them. There are two popular web page ranking algorithms, namely *PageRank* and *HITS* that also are link-analysis algorithms. These algorithms identify important web pages in a web-graph (Huang et al., 2004). *PageRank* method estimates the importance of web pages by constructing an adjacency matrix and assigning a numerical score between *0* and *1* to each page. In this case, different pages are navigated and the importance of each selected page is estimated based on the probabilistic relation between the page with others. The *HITS* algorithm is a recursive algorithm that defines relationships between web pages by employing hubs and authorities. In other words, authority is a page that is linked by many hubs and contains valuable information. Also, hub is a page that links to authorities (Ströele et al., 2013; Ng et al., 2001). Various researches utilize link-analysis algorithms to exploit experts' relationship graph (Xie et al., 2016; Ding et al., 2016). In these researches, experts and their relationships with other experts are defined as nodes and edges of the graph, respectively. Many papers have been provided reports on how to construct experts and their relationships in the social networks (Zhang et al., 2007; Liu et al., 2013; Pal et al., 2011). The following paragraphs describe how two kinds of link-analysis algorithms are applied to expert recommendation systems in order to recommend the appropriate experts.

The idea of language models is borrowed from the field of *NLP*. These models estimate the probability of the occurrence of the next word in a sequence of words by using conditional probabilities. Eq. (7) presents how a language model computes the probability of a sentence or the sequences of words:

$$\begin{aligned} P(w) &= P(w_1 w_2 \cdots w_m) = \prod_{i=1}^{m} P(w_i | w_1 w_2 \cdots w_{i-1}) \\ &= P(w_1) P(w_2|w_1) P(w_3|w_1 w_2) \cdots P(w_m|w_1 \cdots w_{m-1}) \end{aligned} \tag{7}$$

In Eq. (7), if $n = 1$, then *P(w)* is the probability of the next word with the condition of a previous word, called *unigram*. Then if $n = 2$ for the condition of the two preceding words and so on. The most widely used type of statistical language model is the *n-gram* model.

Recently, language Model has become so popular in recommendation systems (Valcarce, 2015; Bonnin et al., 2008). It considers the occurrences of words in the documents and queries as a random generative process (Valcarce et al., 2016). Documents can be ranked according to a user's query by estimating the probability of each document *d* given the topic query *q*, $p(d \mid q)$:

$$p(d \mid q) = \frac{p(q \mid d) p(d)}{p(q)} \overset{rank}{=} p(q \mid d) p(d) \tag{8}$$

where $p(q \mid d)$ is the query likelihood. *p(d)* is the probability of document *d* and $p(q)$ does not have any effect in the ranking for the same queries, hence it can be eliminated. The query likelihood is computed by the *unigram* model based on a multinomial distribution.

$$p(q \mid d) = \prod_{t \in q} p(t \mid d)^{c(t,q)} \tag{9}$$

Here $c(t, q)$ denotes the frequency of term t in the query q. Also, the conditional probability, $p(t|d)$, is the probability of term t in the document *d* (Valcarce et al., 2016).

One of the problems of the language model is the existence of zero in the probability. Smoothing is one of the solutions used to eliminate the zero probability and increase the accuracy (Yang and Zhang, 2010; Harvey et al., 2013; Lin et al., 2017). However, expert recommendation system can utilize above possibilities with some changes. Language-based expert finding systems use probability of Eq. (10) for specifying whether a candidate, *ca*, is an expert for a given query *q* or not:

$$p(ca, q) = \sum_{d \in S} p(ca, q \mid d) p(d) \tag{10}$$

Table 1
Sources and their characteristics.

Source	On-line	Architecture	Quantity	Formal
Quora	Off-line	Structured	Large	Informal
Twitter	On-line	Structured	Large	Informal
Citeseer site	On-line	Structured	Large	Formal
TREC enterprise track	Off-line	Structured	Large	Formal
Yahoo! Answers	On-line	Structured	Large	Informal
Zhihu	Off-line	Structured	Large	Informal
Stack Overflow	Off-line	Structured	Large	Informal
Epinions.com	Off-line	Structured	Large	Informal
Microsoft Office Discussion Groups	On-line	Structured	Large	Informal
DBLP	Off-line	Structured	Large	Formal
Wikipedia	Off-line	Structured	Large	Formal
Google scholar	On-line	Structured	Large	Formal
UvT Expert Collection	Off-line	Structured	Small	Formal
Web pages	On-line	Unstructured	–	Formal
An university	Off-line	Structured	Small	Formal

Table 2
Summarized papers based on the first phase, information retrieval of the experts.

Paper	Source	CBIR	SGBIR	Type
Zhang et al. (2007)	Web pages		✓	Non-QA
Wang et al. (2013)	Microsoft Office Discussion Groups	✓		QA
Zhao et al. (2015)	Quora and Twitter	✓	✓	QA
Elalfy et al. (2018)	Stack Overflow	✓		QA
Neshati et al. (2017)	Stack Overflow	✓		QA
Engin et al. (2014)	Web pages of scholarships			Non-QA
Liu et al. (2013)	Yahoo! Answers – in Taiwan	✓	✓	QA
Davoodi et al. (2013)	Web pages, DBLP, Wikipedia and Google scholar	✓	✓	Non-QA
Xie et al. (2016)	Citeseer site and Twitter	✓	✓	Non-QA
Zhao et al. (2016)	Quora and Twitter	✓	✓	QA
Anongnart (2012)	An university in Thailand	✓		Non-QA
Yang and Zhang (2010)	TREC 2007 enterprise task	✓		Non-QA
Balog et al. (2009)	TREC 2005 and 2006 enterprise track	✓		Non-QA
Liang and de Rijke (2016)	TREC 2005 and 2006 enterprise track	✓		Non-QA
Zhou et al. (2014)	Yahoo! Answers	✓	✓	QA
Yang et al. (2014)	UvT Expert Collection	✓		Non-QA
Zheng et al. (2017)	Stack Overflow and Zhihu	✓	✓	QA
Wang et al. (2017a)	Stack Overflow	✓		QA
Wei et al. (2015)	Epinions.com		✓	QA

where $p(ca, q|d)$ is the conditional probability of candidate expert *ca* and query topic *q* for given document *d*. The candidate expert and the query terms are usually assumed to be independent.

Vector Space Model (VSM) is a formal method for retrieving content that determines the level of relevance of a document to a query. Many studies employ VSM to implement the second stage of the first phase. VSM shows documents and queries in vector spaces and does calculations on them. Various schemes have been developed to compute the values of the vectors, such as Inverse Document Frequency (*IDF*) factor and Term Weighting. Thus, after determining query and document vectors, an *IR* technique is performed based on similarity algorithms. Similarity algorithm such as *Cosine Similarity Measure* calculates demonstrates the similarity between the user's shared items with the topic query and acquires people with the maximum similarity as an expert (Chen et al., 2015a; Beel et al., 2016; Park et al., 2012; Nagarnaik and Thomas, 2015). To give an example, Eq. (11) calculates a cosine similarity between the profile document of each expert's expertise (*d*) and the given query (*q*):

$$sim(d,q) = \frac{\sum_{j=1}^{t} d_j q_j}{\sqrt{\sum_{j=1}^{t} d_j^2 \sum_{j=1}^{t} q_j^2}} \tag{11}$$

where d_j denotes the term frequency-inverse document frequency (TF-IDF) weight for term j in an expertise profile document and q_j denotes the TF-IDF weight for term j in the query vector q (Isinkaye et al., 2015). Another scheme of VSM is a bag-of-words model that is utilized in traditional text analysis. In this model, documents having the most closely matched words with given query, will be found.

Reputation score is one of an expert score that shows his historical activity. Most existing publications represent the relationship of questioner and answerer as a link network and try to find experts in *QAC* without considering the validity of the answers and reputation score. Hence, this survey assumes the reputation score as an approach in expert recommendation systems. An answer with high validity is called best answer. In *QACs*, the best answer is chosen by a questioner to select an answer as the best answer or assigning a vote to each answer and consequently, the answer with the highest vote is chosen as the best answer (Liu et al., 2013). The reputation score is a measure to find that how much the author's answer is trusted (Elalfy et al., 2018).

Another approach is matrix completion approach. Matrix completion is defined as finding the missing values of a matrix given a few of its entries (Kalofolias et al., 2014). In recommendation systems, a matrix completion problem is to find similar rating patterns and use them to complete missing values (Kalofolias et al., 2014).

K-Means is one of the most popular types of clustering algorithms. It is used in many branches of science such as data mining, pattern recognition, information retrieval, and image processing. The aim of this clustering algorithm is to bunch data into disjointed groups in a way that the data in each group be similar (Likas et al., 2003; Žalik, 2008).

For clustering N input data points $x_1, x_2, \ldots, x_N$ into k disjoint groups $C_i, i = 1, \ldots, k$ should minimize the following *Mean-Square-Error* (*MSE*) cost function (12):

$$J_{MSE} = \sum_{i=1}^{k} \sum_{x_t \in C_i} \left\| X_t - C_i \right\|^2 \tag{12}$$

Here $c_1, c_2, c_j, \ldots, c_k$ are called cluster centers which are learned by the following steps: Step 1: Initialize k cluster centers $c_1, c_2, \ldots, c_k$ by some initial values called seed-points using random sampling. For each input data point x_t and all k clusters, steps 2 and 3 are repeated until all centers converge.

Step 2: Cluster membership function $I(x_t, i)$ is calculated by Eq. (13) and then is assigned to each input data point to one of the k clusters whose center is closest to that point.

$$I(x_t, i) = \sum_{i=1}^{k} \sum_{x_t \in C_i} \left\| X_t - C_i \right\|^2 \tag{13}$$

Step 3: For all k cluster centers, c_i is set to be the center of mass of all points in cluster C_i.

In addition, the K-Means clustering algorithm is used in expert recommendation systems in order to categorize experts based on their interests (Likas et al., 2003; Žalik, 2008).

Deep learning is a new approach to solve both supervised and unsupervised learning tasks. The deep architecture models consist of several layers to learn multiple levels of representations and abstractions from data (Zhang et al., 2017). One of the main capacities of deep learning is both determining the most important features and generating new features in the training dataset. This power of deep learning technique allows, particularly, for much better feature extraction from item characteristics in recommendation systems. This provides not only a more accurate modeling but also lead to remarkable improvements in the quality of the recommendations (Karatzoglou and Hidasi, 2017). This part introduces deep learning techniques that have been used in expert recommendation systems to achieve high recommendation quality (Karatzoglou and Hidasi, 2017; Zhang et al., 2017). To give a better view on existing works, this survey organizes current works in two categories: *Convolutional Neural Networks* and *Recurrent Neural Networks*.

Feedforward neural network consists of different layers of neurons organized in the input, output and hidden layers. A fixed number of neurons are considered for each layer. The number of neurons in the input layer represents the number of input features. The neurons in the hidden layer illustrate the number of centroids. The neurons are connected to each other by synaptic weights. Feedforward neural networks process data in a one-way direction and the network weights are adapted until a minimum error is achieved (Ibrahim and El-Amary, 2018). Hence, they are often characterized as being static (Gupta et al., 2016). Convolutional Neural Network(*CNN*) is a subclass of feedforward neural networks that employs convolution and pooling operations in at least one of its layers (Goodfellow et al., 2016). The convolution layer in *CNN* is considered a powerful part for feature extraction. The pooling layers diminish the size of the input in a way that summarizes neurons from a small spatial neighborhood (Scherer et al., 2010). Due to the utilization of *CNN* for feature extraction, there has been a number of notable recommendation systems that employed *CNN* to provide recommendations. Paper Gong and Zhang (2016) is a hashtag recommendation system based on *CNN* model. There is also a *CNN* model for tag recommendation system that is proposed in paper Nguyen et al. (2017). To review more *CNN* based recommendation system, please refer to Zhang et al. (2017). It provides an extensive review on deep learning based recommendation systems.

Recurrent neural network (*RNN*) is a special kind of neural network that processes sequential data (Goodfellow et al., 2016). *RNN* shares the same weights across several time steps. The most effective model of *RNN* is the long short-term memory (*LSTM*) that has a chain like *RNN* but units contain three gates as forget, input and output. These gates control the flow of information through the sequence (Nassif et al., 2016). Additionally, Bidirectional Recurrent Neural Network (*BRNN*), another type of *RNN*, has been designed to use past and future sequences (Nassif et al., 2016). *BRNN* splits the neurons of *RNN* into two directions for processing the sequence forward and backward. *RNN* based approaches have emerged as a promising method to cope with the temporal dynamics of user preferences in recommendation systems (Devooght and Bersini, 2016; Pei et al., 2017). There have been

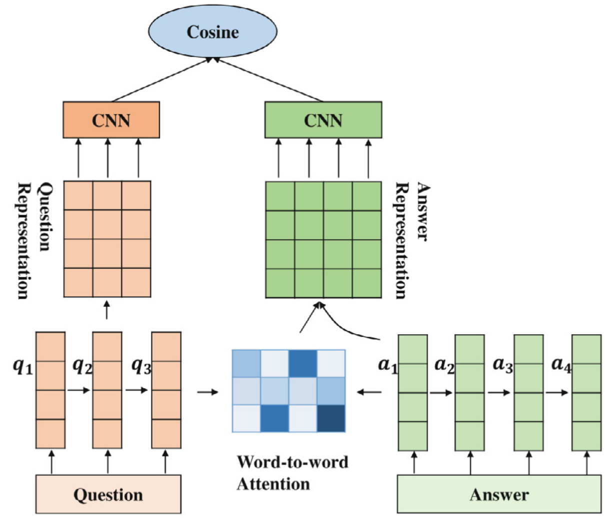

Fig. 6. The architecture of the proposed model in paper Zheng et al. (2018).

many works using *RNN* models for recommendation systems. Lee et al. (2016) advanced a composite model with *RNN* and *CNN* for quote recommendation. Zhang et al. (2017a) proposed a composite model which combines *RNN* and *CNN* for hashtag recommendation (Zhang et al., 2017). *RNN* also has been used a lot for answer selection in question-answering systems. As an illustration, Zheng et al. (2018) proposed a model that integrates *LSTM*, attention mechanism (*ATT*) and *CNN* composite model for question–answer matching. The inputs of *LSTM* network is fed by the question and answer embedding. Moreover, an attention mechanism shows the contribution of each answer word over the question. After that, the *CNN* part generates the representations of question and answer. At the end, the cosine similarity between the question and answer is calculated (Zhou et al., 2018). Fig. 6 shows the architecture of the proposed model.

Nassif et al. (2016) proposed a deep learning model combining *LSTM* and Multi-Layer Perceptron (*MLP*) for question–answer matching. The vector representations of questions and answers are computed with two bidirectional *LSTM*s. As the next step, vectors can be augmented with the additional features. In the next step, vectors with extra features fed into the *MLP* to predict the semantic similarity score of question q and answer a. The architecture of this model is represented in Fig. 7.

Although we can find many studies that apply *RNN* models and combination of deep learning architectures for question-answering matching, but there are very few studies that use these architectures to provide recommendations in expert recommendation systems.

In the following, we attempt to review and analyze the researches that employ the mentioned approaches.

Engin et al. (2014) developed a rule-based expert finding system in order to discover correlations between scholarships and students. The system recommends scholarships to university students according to their eligibility. Furthermore, the rules specify how a specific scholarship is appropriate to a student. The authors have employed *Oracle Policy Automation* (*OPA*). *OPA* is a software that reads and develops rules. It solves a problem based on the list of acquired rules. The conditions of each scholarship are mapped as rules in *OPA*. In order to specify their relevancies to the scholarship, the students select a scholarship and answer some related questions, asked by *OPA*. Finally, the proposed expert system presents the result in one of the forms: “The student is eligible for the special Scholarship” or “The student is not eligible for the special Scholarship”.

Zhang et al. (2007) suggested a propagation-based approach to find experts in a social network. The proposed method computes an initial

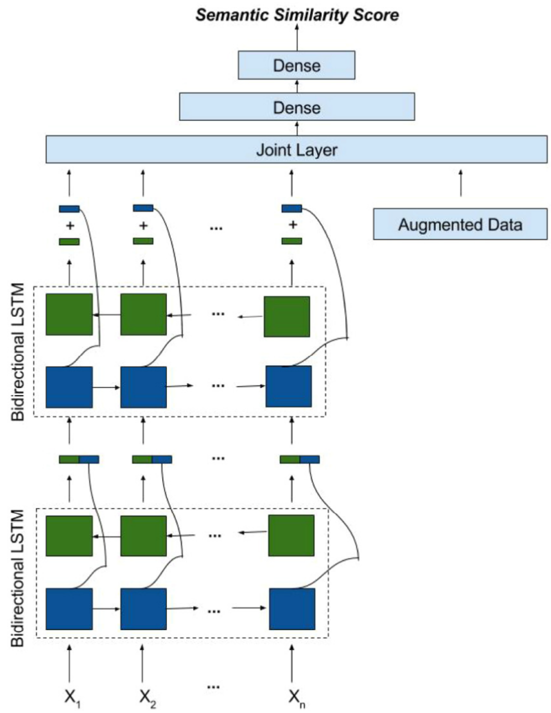

Fig. 7. The architecture of the proposed model in paper Nassif et al. (2016).

score for each expert using the probabilistic information retrieval model and constructs a weighted graph with the experts' initial score. The weight of graph, points out how well the experts' score propagates to their neighbors. The expert score is calculated as Eq. (14):

$$s(v_i)^{n+1} = s(v_i)^n + \sum_{v_j \in U} \sum_{e \in R_{ji}} w((v_j, v_i), e) s(v_j)^n \tag{14}$$

where $e \in R_{ji}$ shows the type of relationship from the expert v_j to v_i; U stands for a set of neighboring nodes to v_i in graph, $w((v_j, v_i), e)$ represents the propagation coefficient and R_{ji} stands for all relationships from the expert v_j to v_i. The above calculation runs in iterations and all expert scores are normalized in each iteration. The propagation is done before normalizing expert scores. The iteration ends when one of the stopping conditions is satisfied; falling the maximal change of the expert score from a predefined threshold, and exceeding the predefined iteration limit (Lin et al., 2017; Zhang et al., 2007).

Authors in Neshati et al. (2017) have paid attention to the temporal and dynamical aspects of the expert finding problem to predict the potential experts in future time. In this study, a language model has been used to estimate the probability $p(e \mid q, CT = t_1, FT = t_2)$ to rank candidate e in future time ($FT = t_2$) while the expertise evidence are given at time $CT = t_1$. This study gathered test collection from *StackOverflow*. It utilized *Lucene Standard Analyzer* to remove the stop-words and to stem the words occurred in the questions and answers. *Lucene* is an open source Java-based search library. *StandardAnalyzer* is the more sophisticated analyzer of *Lucene* that removes common words and punctuations, etc.

Yang and Zhang (2010) made an effort to improve the performance of the expert recommendation system using a language model approach. The authors considered queries and experts as two dependent variables.

$$p(ca, q) = \sum_{d \in S} p_{co}(ca \mid q, d) p_{dep}(d \mid q) p(q) \tag{15}$$

where p_{dep} and p_{co} are calculated by Eqs. (16) and (17), respectively.

$$p_{dep}(d \mid q) = \sum_{d \in S} p_{dep}(q \mid d) \tag{16}$$

$$p_{co}(ca \mid q, d) = \sum_{i=1}^{N} p(w_i) p_{co}(ca \mid q, d, w_i) \tag{17}$$

here $p(w_i)$ is the probability for each of the window-based co-occurrence models, $p_{co}(ca \mid q, d, w_i)$.

$$p_{co}(ca \mid q, d, w_i) = \sum_{w_i} \frac{f(ca, d, q, w_i)}{\sum_{\acute{ca} \, in \, C} f(\acute{ca}, d, q, w_i)} \tag{18}$$

that $\sum_{\acute{ca} \, in \, C} f(\acute{ca}, d, q, w_i)$ is the frequency of all candidates in w_i and $f(ca, d, q, w_i)$ is the number of times that candidate ca co-occur with query topic q in w_i.

Coming to paper Liu et al. (2013), it selected *Yahoo! Answers – in Taiwan* as its dataset to conducted experiments. In this study, an expert score is composed of two parts: knowledge score and authority score. Knowledge score itself consists of two scores. One score considers the relevance of user's subject knowledge to the target question. This score is computed by cosine similarity algorithm. The second score is a reputation score that considers the effect related to quality of users' answers in *QAC*. The reputation score is defined as the ratio of best answers to answers given by the user u_a which is derived by Eq. (19):

$$S^f_{R_{u_a}} = \frac{CategoryAdoptedRatio^f_{u_a}}{\max_{\{u_x\}} CategoryAdoptedRatio^f_{u_x}} \times [\lambda + (1-\lambda) \times \frac{BestAnswer^f_{u_a}}{\max_{\{u_x\}} BestAnswer^f_{u_x}}] \tag{19}$$

The parameter λ, here, adjusts the relative importance of adoption ratio and the number of best answers, $BestAnswer^f_{u_a}$ is the number of best answers in category f given by user u_a. $Category\text{-}AdoptedRatio^f_{u_a}$ is defined as the user u_a's adoption ratio of best answers in category f that is calculated by Eq. (20):

$$CategoryAdoptedRatio^f_{u_a} = \frac{\#best\ answer\ by\ user\ u_a}{\#answer\ by\ user\ u_a} \tag{20}$$

The equation describes that a user will have a higher reputation if he/she have a higher adoption ratio of best answers and had a greater number of best answers. For authority score, the user–user relationship graph is extracted by connecting users who participate in and reply the same question. Then, this research has proposed a method which combined *HITS* and *PageRank* algorithms. Eq. (21) shows the authority score based on *HITS* algorithm that combines the hub score and authority score for each user:

$$H(u_b) = \sum_{u_a : u_b \to u_a} A(u_a); A(u_a) = \sum_{u_b : u_b \to u_a} H(u_b) \tag{21}$$

where H(ub) denotes user's hub value, and $A(u_a)$ denotes user u_a's authority value in *HITS* algorithm. Accordingly, users, who are well at asking questions, have high hub score and the ones with high authority score are considered as good answerers. The authority score based on *PageRank* is calculated by Eq. (22):

$$PR(u_a) = c \sum_{u_b : u_b \to u_a} \frac{PR(u_b)}{O(u_b)} + (1-c)\frac{1}{N} \tag{22}$$

where user u_a's *PageRank* score shows as $PR(u_a)$; $O(u_b)$ denotes user u_b's out degree; c is the damping factor and it is set at *0.85*. N is the total number of users. Finally, a linear combination is merged knowledge and authority scores.

Paper Elalfy et al. (2018) proposed a method based on the reputation score approach. In this work, the reputation score in a specific category is determined by two scores as the confidence score and expertise score. The confidence score is used to certify that users with a high number of best answers have a higher score than others. On the other hands, a user's expertise score is impressed by the user's activity and degree of participation in answering questions. In this way, the expert's reputation score in a specific category is calculated by Eq. (23):

$$S_R(u_{ij}) = con(u_{ij}) expe(u_{ij}) \tag{23}$$

Where $expe(u_{ij})$ and $con(u_{ij})$ denote expertise score and the confidence score, respectively and are computed by Eq. (24) and Eq. (27).

$$expe(u_{ij}) = \frac{f(u_{ij}) + g(u_{ij})}{2} \tag{24}$$

In Eq. (24), function $f(u_{ij})$ measures answerers activeness as Eq. (25) and $g(u_{ij})$ determines the degree of expertise of the user on a specific category and defines by Eq. (26)

$$f(u_{ij}) = \frac{1}{1 + e^{(-\frac{answer\, by\, user\, u_{ij} - \mu}{\sigma})}} \tag{25}$$

$$g(u_{ij}) = \frac{1}{1 + e^{(-\frac{best\, answer\, by\, user\, u_{ij} - \mu_b}{\sigma_b})}} \tag{26}$$

$$con(u_{ij}) = \begin{cases} \dfrac{best\, answer\, by\, user\, u_{ij}}{answer\, by\, user\, u_{ij}} & |\, best\, answer\, by\, user\, u_{ij}\, | > 0 \\ 0.0001 & |\, best\, answer\, by\, user\, u_{ij}\, | = 0 \end{cases} \tag{27}$$

where $answer\, by\, user\, u_{ij}$ is the total number of answers that user u_{ij} provides in category C_i and *m* is the threshold value that is determined based on the answer distribution pattern of the users. Parameter $\sigma = \sqrt{\frac{(answer\, by\, user\, u_{ij} - \overline{answer})^2}{t}}$ and $\overline{answer}$ are the variations over the number of answers and the average number of answers in the category *i*, respectively. Parameter $best\, answer\, by\, user\, u_{ij}$ in the above equations is the total number of best answers that user u_{ij} provides in category C_i and finally, μ_b is the threshold value. Using sigmoid function causes the reputation score to be equal to 1 for large positive numbers and for values greater than the threshold, the score for all users equals one.

Paper Faisal et al. (2018) proposed a novel method for computing reputation score using bibliometric G-index. The paper defines G-index in its proposed approach, Called $Exp - PC$, as a set of posts ranked in decreasing order of the number of posts scores that a user received using Eq. (28).

$$Exp - PC \leq \frac{1}{Exp - PC} \sum_{i \leq Exp - PC} PS_i \tag{28}$$

that *Exp-PC* represents expert rank score and *PS* represents post score. This work illustrates reputation features such as reputation of the voter, ratio of received up-votes to down-votes, a discussion with reputed participants and total popular tags applied by the user. Reputation feature score, called *Rep-FS*, is computed by combining the scores for each of features as Eq. (29):

$$Rep - FS = \sum_{j=1}^{m} \sum_{i=1}^{n} (F_i U_j) \tag{29}$$

where F_i is the mentioned feature score for User U_i. A user's final reputation score, S_R, is measured combination of bibliometric, *Exp-PC*, and reputation feature score, *Rep-FS*, as Eq. (30):

$$S_R = Exp - PC * Rep - FS \tag{30}$$

This study conducted the experiments on *StackOverflow*. The advantage of this reputation score is considering the quality of answers given by users in QAC to find experts.

Researchers in Wang et al. (2013) crawled all on-line posts in *Microsoft Office Discussion Groups* as information source to extract experts' information. After that, they performed stop-word removal and stemming on users' questions and answers to preprocess gathered information. This research has utilized cosine similarity to calculate the relevance score between the posts in online *KC*s that authored by an expert candidate and a topic query. This paper also used *SGBIR* technique to build the Social graph dataset to determine experts' social importance. To extract user–user relationships, a user-thread relationship is constructed; whenever a user initiates a discussion topic, a new thread is created. In this case, a directed edge from a user to a thread denotes the user who has asked the related question. Furthermore, a directed edge from a thread to a user represents the user who has answered the question. Consequently, the relationship between the users can be obtained by the relationship between a thread and its related users. The user–user relationship graph is then built by the links connecting users who asked the question and who give answers to the same thread with directed edges from topic starters to answerers. This work advanced a *PageRank* algorithm, called *ExpertRank* to calculate authority score. For this purpose, an adjacency matrix is created after constructing the user–user relationship graph. The value *1* in the matrix indicates the existence of the relation between two experts and value *0* means that experts do not have any connections with each other. Afterward, the expert's authority score is modified as follow:

$$AU(i) = d \sum_{j=1}^{N} AU(j).u'_{ji} + (1 - d)\frac{1}{N} \tag{31}$$

where *d* is a damping factor similar to that used in *PageRank*, *AU(ca)* denotes an expert's score. u'_{ji} denotes an element in a weighted adjacency matrix U'. Furthermore, the paper proposed a weighted reference relationship *WRR* algorithm for calculating the weights of adjacency matrix. The algorithm consists of three steps where, it constructs a matrix *S* in the first step and after that each element is calculated by Eq. (32).

$$s_{ij} = \begin{cases} 0 & i = j \\ n(i,j) & i \neq j \end{cases} \tag{32}$$

Second step updates the weight as shown in Eq. (33):

$$S'_{ji} = S_{ij} - S_{ji} - \beta \sum_{k} \min(S_{jk}, S_{ki}) - \beta^{m-1} \sum_{k \ldots l} \min(S_{jk}, \ldots, S_{li}) \tag{33}$$

In the last step, each element u'_{ij} is calculated by normalizing S'_{ij}.

$$u'_{ij} = \frac{S'_{ij}}{\sum_{k=1}^{N} S'_{ik}} \tag{34}$$

The novelty of this paper is proposing three strategies to combine expertise relevance and expert authority scores into a single expert ranking score: Linear combination, Cascade ranking, and Scaling strategy. The shortcoming of this paper comes from the recursive nature of topic answering that may cause duplicated extractions of users who are intended to be ranked. Another bothering problem is irresponsible answers that may contain inaccurate or sometimes wrong answers. Better to mention that, users who ask questions in most cases are not professional. Amateur askers and repliers are not detectable in such systems and are yet to be considered as professionals in a topic that they wrongly answered. Anyway, for finding experts in social networks and determining their level of knowledge, the validity of their posts should be considered.

Study Zhao et al. (2015) used *Quora* to conduct its experiments. It crawled the *Quora* to collect the posted questions and all the users who answered these questions between September 2012 and August 2013. Also *Twitter* was used as another source to collect the required information. The users' following relationships were crawled from Twitter graph for the users in *Quora* that have *Twitter* account. Authors in this study have considered the expert recommendation system problem as an illustration of the matrix completion problem. Some notations are introduced in this research which are the data matrix of the questions $Q \in R^{d \times m}$, X matrix of users, W similarity matrix of users. Moreover, the quality of all users in answering the questions is shown as the rating matrix $Y \in R^{m \times n}$. In the data matrix of the questions, each question *q* is presented by a *d*-dimensional word vector using bag-of-words model. To form similarity matrix of users and calculate their similarity, the *Jaccard Distance* is utilized. *Jaccard Distance* is a statistic index for computing the similarity between two sets. The following Equation is used to measure the similarity between two users:

$$W_{ij} = \frac{F_i \cap F_j}{F_i \cup F_j} \tag{35}$$

where F_i is the set of users who follow the *i*th user. The higher value for W_{ij} represents the more similarity for the two users *i* and *j*. In addition, there are a number of missing values in the rating matrix Y. The missing values in this matrix are estimated by using the graph regularized matrix completion and select the users with the highest predicted values as experts. Q, W and Y are given to the system and the goal is to learn expertise function f_x for each user x and complete the missing values in Y. Experts with the highest value of $f_x(q)$ are selected for answering question q. Expertise function f_x is defined by Eq. (36):

$$f_X(Q) = Q^T X \tag{36}$$

Davoodi et al. (2013) selected 315 academic researchers in the field of computer science who were program committee members of the *16th ACM SIGKDD* conference. In the next step, a spider crawled Internet and extracted researchers' information to construct Document dataset. To enhance the user profile completeness and accuracy, they utilized *DBLP* bibliography and retrieved the list of publications related to each researcher. For each publication, its keywords and abstract was retrieved from *Google Scholar*. They also used *Wikipedia* as another source to extract semantic similarities between a pair of words. Authors build a semantic-based profile for each expert based on cosine similarity. The paper proposes to construct a semantic social network of experts based on their profiles and detect communities of experts. Members of communities should be densely connected in terms of expertise, knowledge, and experience. Also, the communities should be weakly connected. The paper employs the K-Means clustering algorithm to detect the communities of experts. In order to minimize the number of clusters (separation), and maximize its quality, (homogeneity), *Euclidean* distance is used to detect the expert communities and calculates the distance between two clusters of experts. The homogeneity shows the density of connections between members of a community and is calculated by Eq. (37):

$$Homogeneity = \frac{1}{C} \sum_{for\ each\ cluster\ K} Hom_k \tag{37}$$

where Hom_k is the homogeneity measurement of the kth cluster in a clustering solution and is calculated by Eq. (38) and C is the number of clusters in a clustering solution.

$$Hom_k = \frac{1}{m} \sum S(i, j) \tag{38}$$

$S(x, y)$ is the semantic similarity between two x and y nodes in the social network. The separation demonstrates less connection between communities and is defined as (39):

$$Separation = \frac{1}{C} \sum_{for\ each\ cluster\ K} Sep_k \tag{39}$$

where Sep_k is the separation of the kth cluster in a clustering solution and is obtained by Eq. (40):

$$Sep_k = \arg \min_{\forall_i \in k, \forall_1 \leq\ n \leq\ C} S(i, j_n) \tag{40}$$

After that, by cosine similarity approach the most similar queries are recommended to community members based on the similarity between the preference of the community representative, and queries. In this paper, the number of clusters is not specified and the aim is to increase the density of clusters that causes high computational costs. However, the usage of *Euclidean distance* is another issue and it is not a desirable metric for high-dimensional data mining applications.

Anongnart (2012) obtained dataset from one of public university in Thailand. Streaming and stop word removal, information preprocessing techniques were applied in document keyword extraction process. Author analyzed experts' preferences based on the clustering experts by K-Means algorithm. In the first step, the center of each cluster is the document with the minimum distance to the clustering features that are extracted from document keywords. The second step consists of two parts. In the first part, each expert is assigned to the cluster based on the distance of the expert's research keywords to the centroid of a cluster. In the next part, a new center for each cluster is obtained. Two parts of the second step run in iterations until there is no change in cluster centers. This paper also considers a threshold for the cluster size because the largest cluster would not have most members of experts.

Zheng et al. (2017) trained their proposed framework on two datasets, *Stack Overflow* and *Zhihu* in order to adapt to different languages. Authors proposed a *CNN* based model that integrated both user and question feature representations. User feature representation is performed with *DeepWalk* method that embeds user's id as 200 dimensions vector. *DeepWalk* (Perozzi et al., 2014a) was proposed by Perozzi et al. that learns latent representations of vertices in a network and maps the social relations into a continues vector space. The question feature is embedded by utilizing *Word2Vec* and *Glove*. After generating question and user vectors, the cosine similarity computes expert's score. In this work, the loss function is defined as a optimization problem using Eq. (41). Loss function evaluates the performance of a system that how well the learner predict the outputs.

$$\min \sum_{u^+, u^-, q} \max[0, margine - \cos(v_{u^+}, h(q, \theta)) - cos(v_{u^-}, h(q, \theta))] \tag{41}$$

where q is the question content, cos means cosine similarity, u^+, u^- denotes two users such that u^+ has higher cosine similarity than u^-. $v_u \in R^d$ is a vector representation of a user u. $h(q, \theta)$ transforms a question q into a d-dimensional vector. θ is the learning parameter. For a question q, an expert user is specified by:

$$userid = argmax_u(\cos(v_u, h(q, \theta))) \tag{42}$$

that u means the *userid*.

The research that is reported in Zhao et al. (2016), made use of two datasets. The first dataset is *Quora* and the second one is *Twitter*. Authors in this work have proposed a heterogeneous network as represented in Fig. 8. This network is a combination of the users' relationship in social networks and their question–answer relationships in *QAC*s. This study uses *DeepWalk* to learn network embedding only from the heterogeneous network. Addition to the heterogeneous network, a *LSTM* network, which is called *Q-LSTM*, is employed to embed the sentences of the questions. If the question is in a paragraph, it will be split into sentences and then the output of *Q-LSTM*s are merged by max-pooling. In the next step, the output of the heterogeneous and *LSTM* networks, which are users and questions embedding vectors, are encoded into fixed feature vectors. After that, these fixed feature vectors are injected as inputs to the ranking metric network. The loss function of this network is described as Eq. (43).

$$L(v_i) = \begin{cases} \sum_{u^+, u^- \in W} max(0, m + f_{u^-}(v_i) - f_{u^+}(v_i)), & v_i \in Q \\ \sum_{u \in W} \|u - v_i\|^2, & v_i \in U \end{cases} \tag{43}$$

where u^+ and u^- denote the high and low quality experts for answering questions, respectively. m is the hyper-parameter that controls the margin in loss function. Q and U are the set of questions and users. W denotes the context windows for network vertices.

Paper Wang et al. (2017a) treated an expert recommendation as a classification problem in *Stack Overflow* question-answering system. It describes the best answerer of a question as positive data and others as negative data. It creates a profile for each candidate expert. The expert profile and questions are presented as word embedding. These vectors fed into the convolutional neural networks to predict best users for answering the newly posted questions. Fig. 9 demonstrated the architecture of *CNN* proposed by this work.

After introducing the categories and expressing previous studies, it turns to review the advantages and disadvantages of each category. The results that are obtained by rule-based systems, are the most reliable accurate results. Due to predefined rules here, the error rate is less. Moreover, these systems are cost-efficient. On the other hand, there are some challenges for the rule-based systems. These systems are expensive to train. Also, in recommendation systems, it is important to update

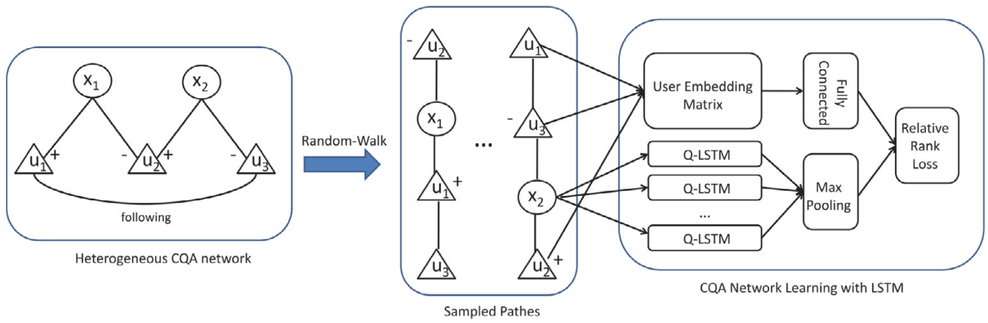

Fig. 8. The Heterogeneous Network proposed in paper Zhao et al. (2016).

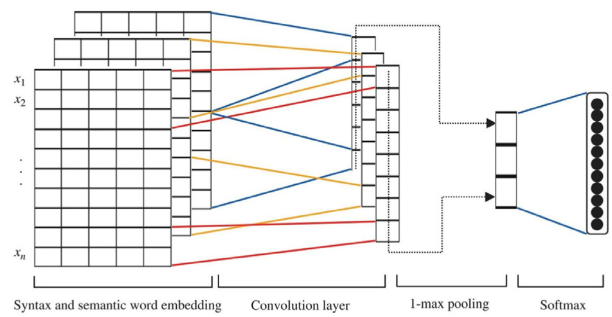

Fig. 9. The architecture of *CNN* used for expert recommendation in paper Wang et al. (2017a).

the user's model, frequently, in order to generate new recommendations instantly (Grosan and Abraham, 2011). Less learning capacity and complex pattern discovering are other challenges for these systems. For example in paper Engin et al. (2014), if the rules of scholarships change, rule-based models have to be retrained. This, consequently, demands a lot of manual work and is time-consuming.

Although that propagation technique enables us to handle finding the top-ranked experts (by setting a relevance threshold), it may miss some important structural aspects. Furthermore, propagation function is time-consuming and produces redundant score messages. As a node is able to receive multiple score messages and it causes a heavy overhead.

The key functionality of link-analysis is its ability to organize the data in the form of a graph. Explicitly, the graph representation makes it easier to present information which is too complicated to be described by text. To analysis experts' activities, link analysis approach utilizes useful knowledge obtained from the relationships in social networks. There are some limitations in the usage of link analysis approaches; namely, certain number of posted questions have no any answers. Hence, in the user–user relationship graph, there are some nodes that do not have any out-links. In *PageRank* algorithms, this is called dangling nodes. The number of users and questions are increased daily in these networks. Using a static graph to represent the user–user relationship causes not to consider new questions and users who may be experts in some knowledge areas. The fact that link analysis requires an extensive amount of data preparation, will poses some expenses to dynamic user–user graph building.

Language model provides a solution to solve the critical issue of text representation and term weighting. They are applicable even using small corpora. Despite its benefits, the language model has some drawbacks, including computation complexity and high time-consumption. It is infeasible to handle large corpus because of the huge amount of computations.

In spite of the fact that VSM approaches are simple in implementation, they do not consider semantic similarities among words and disregards the relationship among terms (Davoodi et al., 2013). For example it will assume two documents containing just one word, that are different in appearance but semantically similar, completely disjointed/different. Suppose that the words Smart and Clever, are used in two different documents. Although they are synonyms, the documents will have cosine distance. TF-IDF, also suffers another disadvantage as shorter documents have a better matching with query terms.

Most of the existing reputation score approaches apply features such as the total number of answers and the number of best answers to calculate reputation score. These studies ignore the quality and the consistency of the user's answer (Faisal et al., 2018). The best way to calculate the reputation score is considering the quality of answers given by users in *QAC*. Many studies show that there is useful information in online *QAC*s with good features to make question-answering systems which generate answers of questions and identify valid and best answers (Liu et al., 2010; Minaee and Liu, 2017; Zheng et al., 2017). The efficiency of such systems can be increased by combining them with expert finding systems. In the learning phase of these systems, the answers of the expert with the highest reputation score help to predict the best answers for new questions. After that, the system can automatically find best answers for new questions with no need to answerers. The fundamental issue is that varying the quality of answers, determining the best answer will take more efforts and will be time-consuming.

In expert finding problem in *QAC*s, this approach tries to estimate missing values in the rating matrix. The matrix of questions and users are n-dimensional because there are a large number of users and questions in these environments. As a result, calculating the determinant of question matrix as well as the multiplication of matrices has huge computational complexity.

Generally, K-Means approach for finding experts has some drawbacks including high computational during computing similarity and high overhead during clustering (Surya Narayana and Vasumathi, 2017). Moreover, it is difficult to find the value of k. Besides finding an optimal k, there is also another problem. Datasets are not fixed and they may change over the time, so k should not be a static number.

Figs. 9–8 are all from the reviewed studies to represent which architectures of deep approaches are employed by current publications and what parts of these architectures can be improved by future studies. Numerous methods have been proposed based on deep learning for question–answer matching. Question–answer matching is not the main focus of this paper, that is why we have just mentioned a few of these researches in this survey. As outlined in the figures, there are very few studies based on deep learning approaches to find experts that we have pointed out in this survey. Deep learning can understand the content of the question appropriately and can learn user's expertise in each domain that causes to improve the quality of recommendation and consequently it increases performances. Despite the benefits of deep learning approach, the basic problem of this approach in expert recommendation systems is the lack of suitable data and dataset for training the neural network and this causes that many studies are limited to find experts in question-answering systems. Another issue related to deep approaches is that multiple layers introduce complex error space. This means that a lot of arguments have to be tuned in order to understand why the system has reached a certain conclusion. Although the mentioned problems are valid for all deep learning structures, but each structure has its own disadvantages. A major problem of using *CNN* is it's sensitivity to the text size. This problem is in some cases handled with padding 0 rows at the end of matrices to make them have the same size. Although padding improves performance by keeping information at the borders, but this solution may effect on the results. For example, query size is too shorter than document, in this way padding causes more similarity between query and document. Ont the other hand, LSTM works very well for some problems, but some of the drawbacks are that this structure easily overfits. Moreover, it takes longer and require more memory to train.

We summarize the content of this section in two tables. Table 3 demonstrates the advantages and disadvantages of different approaches. Table 4 lists all of the reviewed publication that is categorized based on approaches.

5. Evaluation

We analyzed the current literature and found in order to evaluate the performance of the current expert recommendation systems, ground truth data is required. This is what a supervised learning model does. In other words, the current expert recommendation systems are based on supervised learning and the predicted output by a recommendation model is compared with the corresponding ground truth. The result of the comparison, called the error, provides feedback to the learner. This allows the learner to refine itself to make more accurate recommends.

There are numerous metrics in order to evaluate the quality and performance of expert recommendation systems (Isinkaye et al., 2015). This section provides an overview of evaluation metrics that have been performed to show the usefulness of the expert recommendation systems.

- Precision
  Precision determines the fraction of recommended experts who their expertise are fully relevant to the query topic (Isinkaye et al., 2015). In other hands, it is defined as the fraction of the predicted list Y' (true positive) out of all predicted experts (true positive and false positive) (Wang et al., 2013; Shani and Gunawardana, 2011; Aggarwal, 2016; Wu et al., 2012; Gunawardana and Shani, 2015). As clearly explained in Eq. (44) as precision is the proportion of recommended experts that are actually good.

$$Precision = \frac{|Y' \cap Y|}{|Y'|} \tag{44}$$

  Precision at K, $P@K$, is the fraction of the *top-K* retrieved experts that are relevant to the question/query.
- Recall
  Recall shows the fraction of relevant experts that are also recommended (Wang et al., 2013; Shani and Gunawardana, 2011; Aggarwal, 2016; Wu et al., 2012; Gunawardana and Shani, 2015). It measures the proportion of all good experts recommended as Eq. (45):

$$Recall = \frac{|Y' \cap Y|}{|Y|} \tag{45}$$

- F-measure
  The ***F-measure*** of the expert recommendation system is defined as the weighted harmonic mean of its precision and recall (Zhang and Zhang, 2009). This metric is a combination of precision and recall and considers them into a single metric (Isinkaye et al., 2015). The F-measures value will be high, if both recall and precision be high. Whenever F-measure is defined as Eq. (46), it will be known as F_1-measure. In this case, F_1-measure gives equal weight to precision and recall (Aggarwal, 2016; Powers, 2011).

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{46}$$

- Mean Average Precision (MAP)
  This is a metric that averages the precisions among different queries. This metrics is useful for validating a system's precision among different query types and is expressed by Eq. (47) (Manning et al., 2008; Liu et al., 2013; Zhang and Zhang, 2009; Mogotsi, 2010). Q and $\overline{Precision}(q)$ are number of queries and average of precision over query q, respectively.

$$Mean\ Average\ Precision = \frac{\sum_{q=1}^{Q} \overline{Precision}(q)}{Q} \tag{47}$$

- Root Mean Square Error (RMSE)
  Predicted ratings and actual ratings can be compared by two metrics of *Mean Absolute Error* (*MAE*) and *Root Mean Square Error*. Both of these metrics emphasis on difference of system and actual ratings for experts. These metrics are expressed by Eqs. (48) and (49). In both of the relations, y'_i denotes to predicted ranking and y_i shows the actual value.

$$Mean\ Absolute\ Error = \frac{1}{|Y|} \sum_{i=1}^{|Y|} |y'_i - y_i| \tag{48}$$

$$Root\ Mean\ Square\ Error = \sqrt{\frac{1}{|Y|} \sum_{i=1}^{|Y|} (y'_i - y_i)^2} \tag{49}$$

- Discounted Cumulative Gain (DCG)
  This metric measures usefulness, or gain of experts in the list that should be shown or extracted. Moreover, logarithmic reduction of grading relevance value is used in this metric. The idea behind it, is to penalize highly relevant experts in the list, in order to further correction of the list that is extracted. This metric is formally used by search engines and in their query pages in which many results are shown. This metric is described in Eq. (50).

$$Discounted\ Cumulative\ Gain = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2(i+1)} \tag{50}$$

  Here, p is a particular rank position and rel_i is the graded relevance of result at position i.

We formerly reviewed important metrics used in expert recommendation systems. In addition to the mentioned parameters, each paper may use different parameters to evaluate the performance of its proposed algorithm. For example, user eligibility is the parameter that was evaluated in Engin et al. (2014) and Tung et al. (2010) in average query times by the efficiency of system. Table 5 lists publications based on different evaluation metrics and ground truth that are used to determine the performance of their proposed approaches.

Table 3
Summarized advantages and disadvantages approaches used in the second phase, predicting expertise.

Categories	Advantage	Disadvantage
Rule based model	Accurate results, less error rate, cost-efficient	Less learning capacity, complex pattern discovering, need for manual changes
Propagation	Handling finding the top ranked experts by setting a relevance threshold	Time-consuming, a heavy overhead because of redundant score messages
Link analysis	Modeling users' activities and their relationships in social network	The cost of building dynamic user–user graph, an extensive amount of data preparation
Language model	Solving the issue of text representation and term weighting, working with little corpus data	Computational complexity
K-Means	Easy to implement, grouping experts	Difficult to predict the number of clusters, sensitive to scale
Matrix completion	Estimating missing values in the rating matrix	High computational cost for the determinant matrix calculation
Vector Space Model	Simple in implementation, normalizing the resulting probabilities	Not considering semantic similarities, the relationship between terms
Reputation score	Considering the quality of user's answers in ranking experts	More efforts needed to determine the best answer
Deep learning	Automatic feature extraction, good accuracy in question answering matching	High computational cost of training, complex error space

Table 4
Summarized approaches based on reviewed publication.

Categories	Publications
Rule based model	Engin et al. (2014) and Tung et al. (2010)
Propagation	Zhang et al. (2007) and Serdyukov et al. (2008)
Link analysis model	Wang et al. (2013), Liu et al. (2013), Xie et al. (2016), Kardan et al. (2011), Zhou et al. (2014) and Wei et al. (2015)
Language model	Xie et al. (2016), Yang and Zhang (2010), Balog et al. (2009), Liang and de Rijke (2016), Neshati et al. (2017) and Zhou et al. (2014)
K-Means	Davoodi et al. (2013) and Anongnart (2012)
Matrix completion	Zhao et al. (2015)
Vector Space Model	Wang et al. (2013), Liu et al. (2013), Davoodi et al. (2013), Yang et al. (2014) and Zheng et al. (2017)
Reputation score	Elalfy et al. (2018), Faisal et al. (2018), Liu et al. (2013) and Zhou et al. (2014)
Deep learning	Zhao et al. (2016), Zheng et al. (2017) and Wang et al. (2017a)

Table 5
Ground truth and different evaluation metrics.

Paper	Ground truth	Evaluation metric
Zhang et al. (2007)	–	MAP, P@5, P@10, P@20, P@30, R-Precision, bpref
Wang et al. (2013)	Microsoft official member competence lists	Precision, Recall, Macro-Average F-measure, Micro-Average F-measure, P@10, P@20
Zhao et al. (2015)	Answerers and their received thumbs-up/down	MRR, NDCG, P@1
Elalfy et al. (2018)	Academic category of Stack Overflow	Precision, Recall
Neshati et al. (2017)	Answerers with more than 10 accepted answers	MAP, P@1, P@5, P@10
Engin et al. (2014)	–	User eligibility
Liu et al. (2013)	Three human raters justify manually the suitable experts	MRR, P@5, MAP
Davoodi et al. (2013)	Questionnaires designed for different researchers	P@1, P@3, P@5
Xie et al. (2016)	Users whose number of followers ranks top k	MAP, NDCG@5, NDCG@10
Zhao et al. (2016)	Answerers and their received thumbs-up/down	NDCG, P@1
Anongnart (2012)	–	RMSE and RS (RSquare)
Yang and Zhang (2010)	Ground truth of TREC 2007 expert finding task	MAP
Balog et al. (2009)	Members of the corresponding working group	MAP, Mean Reciprocal Rank(MRR)
Liang and de Rijke (2016)	Constructing three ground truth by using the ground truth of TREC 2005 and 2006 expert finding task	NDCG, NDCG@5, NDCG@10, MAP, P@5, P@10
Zhou et al. (2014)	Average score from two human judges	normalized Expert Quality Measure (nEQM), MAP
Yang et al. (2014)	–	P@5, P@10, P@15, P@20
Zheng et al. (2017)	Users receive a lot of agree number	–
Wang et al. (2017a)	Users with the best answer	Success-at-N
Wei et al. (2015)	Top reviewers and advisors selected and recognized by Epinions.com	Recall 20-100, P@20-100

Table 6
MAP metric on different papers with different datasets.

Paper	MAP	NQ	NA	NU	ND	NT
Zhang et al. (2007)	0.1103	–	–	1,781	–	13
Neshati et al. (2017)	0.697	810,071	1,510,812	Quesr:270,972 Auser:206397	–	50
Liu et al. (2013)	0.44	52,899	215,504	136	–	4
Xie et al. (2016)	0.829	–	–	–	1000	–
Yang and Zhang (2010)	0.494	–	–	–	–	–
Balog et al. (2009)	0.2053 0.4660	–	–	–	331,037	TREC 2005: 50 TREC 2006: 49
Liang and de Rijke (2016)	0.7365	–	–	–	331,037	TREC 2005: 50 TREC 2006: 49
Zhou et al. (2014)	0.543 0.512	84,589 190,432	236,107 413,568	6,872 22,345	–	3

Table 7
Precision@5 metric on different papers with different datasets.

Paper	Precision@5	NQ	NA	NU	ND	NT
Zhang et al. (2007)	0.6154	–	–	1,781	–	13
Neshati et al. (2017)	0.679	810,071	1,510,812	Quer:270,972 Auer:206,397	–	50
Liu et al. (2013)	0.54	52,899	215,504	136	–	4
Davoodi et al. (2013)	0.784	–	–	315	–	–
Liang and de Rijke (2016)	0.8	–	–	–	331,037	TREC 2005: 50 TREC 2006: 49
Yang et al. (2014)	0.91	–	–	2,876	10,479	–

Table 8
Precision Metric on different papers with different datasets.

Paper	Precision	NQ	NA	NU	ND	NT
Wang et al. (2013)	0.3934	228,787	624,219	121,289	–	19
Elalfy et al. (2018)	Random forest:0.711 Logistic regression:0.542 Naïve Bayes:0.542	–	18,158	–	–	–
Wei et al. (2015)	books:0.5072 music:0.4217	–	–	books:12,063 music:10,645	–	–

Table 9
Recall Metric on different papers with different datasets.

Paper	Recall	NQ	NA	NU	ND	NT
Wang et al. (2013)	0.3018	228,787	624,219	121,289	–	19
Elalfy et al. (2018)	Random forest:0.741 Logistic regression:0.736 Naïve Bayes:0.736	–	18,158	–	–	–
Wei et al. (2015)	books:0.3528 music:0.4952	–	–	books:12,063 music:10,645	–	–

Table 10
NDCG Metric on different papers with different datasets.

Paper	NDCG	NQ	NA	NU	ND	NT
Zhao et al. (2015)	0.965	444,138	887,771	95,915	–	–
Zhao et al. (2016)	0.741	444,138	887,771	95,915	–	–
Liang and de Rijke (2016)	0.8949	–	–	–	331,037	TREC 2005:50 TREC 2006: 49

6. Comparison

In order to understand how difficult the task of finding experts is, we illustrate the state-of-the-art experimental results in benchmark datasets in Tables 6–10.

There are some problems to discuss the experimental results. Accordingly, we considered different ways to categorize and analyze the results. At the first idea, we intended to look at the results obtained from different papers on the same datasets, but using different evaluation metrics made it impossible to compare results. In the second method, our goal was to evaluate results based on different approaches that were used by different papers. However, we faced the same problem in first idea. Finally, we decided to evaluate papers based on their common metrics. For example, in Table 6, we compared papers which used *MAP* metric in their measurements. The problem of this analysis is that this comparison has not been done in the same conditions, *e.g.* datasets with the same features. The main purpose of this type of evaluation is to show the best obtained value for each metric.

For a better comparison of the parameters obtained from different methods, some information of their datasets are presented in all tables. The mentioned information has the most important effect on the parameter value such as Number of Questions (*NQ*), Number of Answers (*NA*), Number of Users (*NU*), Number of Documents (*ND*) and Number of Topics (*NT*).

Table 6 shows the comparison of the *MAP* value for papers that have evaluated this parameter in their works. We can see that the maximum value for *MAP* is *0.79* that is related to paper Xie et al. (2016). The authors used *Text similarity* score and *Authority* score for each expert. *Text similarity* score was calculated by a language model approach. Moreover, the link analysis approach, *HITS*, measured *Authority* score in the mentioned research. To point out, this value of *MAP* has been achieved on *Citeseer* site and *Twitter* datasets. It is also interesting to mentioned that paper Liang and de Rijke (2016), with *MAP* value 0.7365, is based on the language model approach too.

Table 7 represents the achieved values of *Precision@5* for papers which have considered this metric in the experiments. As it is illustrated, the highest value of *Precision@5* is happened in Yang et al. (2014). In this research the initial query is first extended and then cosine similarity approach is applied for computing *Text similarity* score. Furthermore, the second level for the value of the *Precision@5* is estimated by Liang and de Rijke (2016). This study was based on the language model approach that measured *Text similarity* score for each expert.

Tables 8 and 9 demonstrates precision and recall, respectively. Paper Elalfy et al. (2018) is markedly achieved better precision (0.711) and recall (0.741) than others. This paper is one of studies that calculated the *Reputation* score for each expert. From the presented tables, it can be inferred explicitly that precision and recall values are very low in expert recommendation systems. Thereupon, existing expert recommendation systems do not accurately recommend experts who their expertise are fully relevant to the query topic.

The results shown in Table 10 are the comparison of different papers based on *NDCG* value. Clearly, paper Zhao et al. (2015) obtained maximum *NDCG* value. This study was based on matrix completion approach that found experts in question-answering community. One can easily notice that the value related to paper Liang and de Rijke (2016) is in the second place.

7. Discussion

Although good expert recommendation systems have been proposed, but they face some challenges and need to be improved. Some of these challenges are failure to using multimedia sources, failure to extend expert recommendation systems to other areas, usage of traditional text information retrieval, failure to employ recent techniques for graph analytics to calculate reputation and authority scores, only a few studies developed deep learning models for expert finding and the absence of appropriate strategies to combine scores and calculate the final score y_i'. Here are some strategies that work for the mentioned challenges and can be effective for the future evolution of expert recommendation systems.

One of the challenges is that experts' expertise is limited to two categories of information: textual and social relationships. While, the usage for multimedia data such as image, video, and audio is increasing with technological advances in storage and communication. So, mining knowledge from multimedia information becomes more important. The solution is to fully and effectively use multimedia sources which contain valuable information about experts' expertise. For example, *YouTube* is a multimedia source. Many experts share videos of their training classes or workshops on *YouTube*. A major benefit of these videos is expressing expert's creativity by analyzing these videos. Furthermore, experts share their achievements, comments and opinions on *Twitter* to inform subscribers about their views and thus, this is another helpful source that can be applied to calculate *Authority* score. Utilizing the mentioned popular resources provides a situation to accurately recognize an expert's expertise areas. It should be underlined that although getting access to multimedia data is easy, extracting information from these sources is also challenging.

By review of existing researches, it can be concluded that most studies focus on finding the experts in *QAC*s. Although, expert recommendation systems can be extended in a variety of domains like healthy, e-commerce and other areas that mentioned in Section 3. For example in e-commerce, the experts are consumers who can offer advantageous reviews and have more knowledge about the products. This helps other costumers that do not spend more time to gather more information about products and to decide whether to buy. Finding experts and creating an expert team provide a situation that can be leveraged in order to achieve a common goal in projects. According to this viewpoint, expert recommendation systems are applicable to a wide range of tasks in civil engineering, mechanical engineering, electrical engineering and other domains to make a group of experts. After forming the desired experts, expert systems can be designed based on these experts' observations and experience to do tasks such as prediction, modeling, diagnosis, investigation and etc.

As reviewed, textual sources are commonly used in all expert recommendation systems. Another drawback of the current works is that they have used traditional text information retrieval. These techniques count frequencies of the query terms in the document content without paying attention to semantic relations between query and document. The way forward could be to employ deep learning approaches that provide new opportunities for information retrieval and extract query and document understanding. These approaches represent query and document as multiple feature vectors in order to show the meaning of sentences in query and document. In order to embed context information, embedding techniques such as *Word2Vec*, a two-layer neural network, Mitra and Craswell (2017) are proposed (Karatzoglou and Hidasi, 2017). As another solution, Mohan et al. (2018) proposed a deep learning approach to modeling the relevance of text in a document to a query. In this work, a variable-length difference vector has computed between the query and document. Many existing models use *Siamese* architecture, especially for short text matching. That employs a unique structure to learn the similarity between inputs (Koch et al., 2015). Correspondingly, the *Deep Semantic Similarity Model* (DSSM) is a special type of *Siamese* architecture that trains on query and document title pairs where both the pieces of texts are represented as bags-of-character-trigraphs (Mitra and Craswell, 2017). It can be inferred that deep learning approaches have the capability of accurately learning distributed representations of natural language expressions like sentences, to improve the performance of content information retrieval techniques (Li and Lu, 2016).

Another challenge is that most researches have taken into consideration the question–answer relationship and employed link analysis techniques to retrieve social information of experts. With the expansion of online social networks, considering experts' relationships provides a strong evidence for experts to have a common background in various online social networks, such as *Facebook* or *Twitter* (Zhao et al., 2016). To analyze experts' activities in social networks, computing reputation and authority scores, the connection between experts are often represented as graphs. Moreover, community structure is one of the most relevant features of graphs and community detection is a key tool for finding the community structure (Fortunato, 2010). In the expert recommendation system, the knowledge of community structures gives a better understanding of the expert status within a group and the relationships between him/her and his/hers neighbors. Communities of experts can be found by measuring the similarity of experts' attributes. Expert's attributes have defined the features such as one's research topics, the categories that they are members, and so on. Analyzing these communities yields insight into different patterns of expert's activities. Recently, many studies have emerged to extend deep learning approaches for community detection (Cavallari et al., 2017). For example graph embedding, as a neural network tool to present graph nodes to low-dimensional feature vectors, helps improve the accuracy of scores. DeepWalk (Perozzi et al., 2014b), LINE (Tang et al., 2015) and node2vec (Grover and Leskovec, 2016) are samples of the existing graph embedding approaches. Also, the community detection task can merge with other approaches such as affective computing and sentiment analysis (Cambria, 2016). Affective computing and sentiment analysis provides a solution for retrieving users' relationships in social networks. It assists to find communities of experts who are particularly interested in the same topics. A key task of affective computing and sentiment analysis, and opinion mining is aspect extraction (Wang et al., 2016). In *Twitter*, an aspect extraction can extract people's opinions from tweets. On the other hand, it can retrieve user–user relationships based on opinions and create communities of experts with similar opinions. The review of proposed techniques for opinion mining shows that combining aspect extraction for opinion mining with deep *CNN*, results in better accuracy (Poria et al., 2016). It is obvious that using opinion mining with deep convolutional neural networks can improve techniques to detect communities of experts with similar opinion and analyze their social activities.

An expert model contains primarily of knowledge about the expert's preferences and the learner element is responsible to model these preferences and compute scores. Moreover, deep comprehension of experts and their expertise are essential in order to accurately make recommendations. Deep neural networks provide more powerful tools for learning the characteristics of the experts and bring more chances to improve the performance of recommendation (Wang et al., 2018). Although many methods have been proposed based on deep learning models such as *CNN*, *RNN*, *LSTM*, *NLP* and so on for the question and answer matching systems, there have already been a few studies which utilize these models or combined them together in order to enhance the performances of expert recommendation systems. Autoencoder, one of commonly used deep learning models, can be extended to the expert recommendation system. In this model, the number of neurons in the input layer are the same as the number of neurons in the output layer. Hence, it can learn the most important features of the input data and reconstruct the original input. In recommendation task, it enables the system to learn lower-dimensional feature representations from experts at the middle-most layer. *RNN* is suitable for tackling with the expert temporal dynamics of features and sequential patterns of expert behaviors (Wang et al., 2018). Therefore, expert recommendation system based on *RNN* can effectively memorize the expert historical sequence behavior and learn its effect of on the expert current behavior. In addition, attention mechanism is popular because of its ability to interpret and visualize what the model is doing. An attention mechanism decides which part of the input should be selected to generate the next output (Seo et al., 2017). This mechanism has been successfully integrated with *MLP*, *RNN*, *CNN*, *LSTM* and other deep neural networks and has achieved remarkable results. Integrating attention mechanism into *RNN*s provides a better memorization of inputs. Combining the attention mechanism with *LSTM* enables the expert recommendation system to model the changes of expert preferences over time (Wang et al., 2017c; Li et al., 2017). Moreover, attention mechanism with *NLP* has the ability to encode long term expert's contextual information and capture hierarchical patterns of these information (Zhao and Wu, 2016).

Although only a few attempts have been made to consider all three scores (*Text similarity*, *Reputation* and *Authority*), but it is important how the learner integrates these scores and computes an expert' score, y_i'. Most of the existing works such as (Liu et al., 2013) and Zhou et al. (2014) use a linear combination; while, Wang et al. (2013) and Elalfy et al. (2018) proposed a nonlinear combination of scores. Each score has a different priority depending on the usage of expert recommendation systems. To give an instance, for finding experts in *QAC*s, *Reputation* score is more important. Therefore, a nonlinear combination may improve the performance of expert recommendation systems. On the other hand, it is hard to determine the coefficients of a nonlinear combination. One solution is that the expert recommendation system can be observed as a multi-objective optimization problem. The Pareto-based approach is one of the approaches to solve a multi-objective optimization problem. The Pareto-based approach treats the users' three scores as separate objective functions that should be maximized. The output of a Pareto-based approach is a set of non-dominated solutions. Hence, it is necessary to employ a decision maker to select the best solution among the set of generated non-dominated solutions (Reihanian et al., 2017).

8. Conclusion

Expert recommendation systems provide an opportunity to identify the most experienced people in each area of the knowledge. This paper thoroughly reviewed the literature on expert recommendation systems. Indeed, researchers and educators who are interested in expert recommendation system, can use this survey. The current survey:

- proposed a procedure for an expert recommendation system, including two main phases, namely information retrieval of the expert and prediction of expert's level of expertise.
- provided an overview of the state-of-the-art expert recommendation systems and highlighted their advantages and disadvantages.
- touched various evaluation metrics.
- enumerated the existing challenges and outlooked promising directions for future research.

Acknowledgment

This work is supported by University of Tabriz, Iran, grant number S/819.

References

Aggarwal, C.C., 2016. Evaluating Recommender Systems. Springer International Publishing, Cham, pp. 225–254.

Ai, P., Xiao, Y., Zhu, K., Wang, H., Hsu, C.-H., 2015. A Personalized News Recommendation System Based on Tag Dependency Graph. Springer International Publishing, Cham, pp. 584–586.

Akram, M., Rahman, I.A., Memon, I., 2014. A review on expert system and its applications in civil engineering. Int. J. Civ. Eng. Built Environ. 1.

Al-Nazer, A., Helmy, T., Al-Mulhem, M., 2014. User's profile ontology-based semantic framework for personalized food and nutrition recommendation. Procedia Comput. Sci. 32, 101–108.

Al-Taie, M.Z., Kadry, S., Obasa, A.I., 2018. Understanding expert finding systems: domains and techniques. Soc. Netw. Anal. Min. 8, 57.

Alarfaj, F., Kruschwitz, U., Hunter, D., Fox, C., 2012. Finding the right supervisor: Expert-finding in a university domain. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, NAACL HLT '12. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1–6, http://dl.acm.org/citation.cfm?id=2385736.2385738.

Alison, C., 1997. The Essence of Artificial Intelligence. Pearson Education India.

Alyguliev, R.M., 2007. Analysis of hyperlinks and the ant algorithm for calculating the ranks of Web pages. Autom. Control Comput. Sci. 41, 44–53.

Anongnart, S., 2012. Building expert: System for searching experts in research university using K-MEANS algorithms. In: Computers & Informatics, ISCI, 2012 IEEE Symposium on. IEEE, pp. 176–179.

Balog, K., Azzopardi, L., de Rijke, M., 2009. A language modeling framework for expert finding. Inf. Process. Manage. 45, 1–19.

Balog, K., Fang, Y., de Rijke, M., Serdyukov, P., Si, L., 2012. Expertise retrieval. Found. Trends Inf. Retr. 6, 127–256.

Balog, K., Thomas, P., Craswell, N., Soboroff, I., Bailey, P., De Vries, A.P., 2008. Overview of the TREC 2008 enterprise track. Technical Report. Amsterdam Univ (Netherlands).

Bayati, S., 2016. Security expert recommender in software engineering. In: Proceedings of the 38th International Conference on Software Engineering Companion, ICSE '16. ACM, pp. 719–721.

Beel, J., Gipp, B., Langer, S., Breitinger, C., 2016. Research-paper recommender systems: a literature survey. Int. J. Digit. Lib. 17, 305–338.

Beemer, B.A., Gregg, D.G., 2008. Advisory systems to support decision making. In: Handbook on Decision Support Systems 1. Springer, pp. 511–527.

Bhat, M., Shumaiev, K., Koch, K., Hohenstein, U., Biesdorf, A., Matthes, F., 2018. An expert recommendation system for design decision making: Who should be involved in making a design decision? In: 2018 IEEE International Conference on Software Architecture, ICSA, pp. 85–8509, http://dx.doi.org/10.1109/ICSA.2018.00018.

Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A., 2013. Recommender systems survey. Knowl.-Based Syst. 46, 109–132.

Bonnin, G., Brun, A., Boyer, A., 2008. Collaborative filtering inspired from language modeling. In: Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the. IEEE, pp. 192–197.

Boratto, L., Carta, S., Fenu, G., Saia, R., 2017. Semantics-aware content-based recommender systems: Design and architecture guidelines. Neurocomputing 254, 79–85, Recent Advances in Semantic Computing and Personalization.

Cambria, E., 2016. Affective computing and sentiment analysis. IEEE Intell. Syst. 31, 102–107.

Cao, L., 2016. Non-IID recommender systems: A review and framework of recommendation paradigm shifting. Engineering 2, 212–224.

Cavallari, S., Zheng, V.W., Cai, H., Chang, K.C.-C., Cambria, E., 2017. Learning community embedding with community detection and node embedding on graphs. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17. ACM, New York, NY, USA, pp. 377–386. http://dx.doi.org/10.1145/3132847.3132925, http://doi.acm.org/10.1145/3132847.3132925.

Chandak, M., Girase, S., Mukhopadhyay, D., 2015. Introducing hybrid technique for optimization of book recommender system. Procedia Comput. Sci. 45, 23–31, International Conference on Advanced Computing Technologies and Applications (ICACTA).

Chang, N., Irvan, M., Terano, T., 2013. A TV program recommender framework. Procedia Comput. Sci. 22, 561–570, 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013.

Chen, L., Chen, G., Wang, F., 2015a. Recommender systems based on user reviews: the state of the art. User Model. User-Adapted Interact. 25, 99–154.

Chen, H., Ororbia II, A.G., Giles, C.L., 2015b. ExpertSeer: a keyphrase based expert recommender for digital libraries. CoRR abs/1511.02058.

Chen, Z., Wu, D., Lu, J., Chen, Y., 2013. Metadata-based information resource integration for research management. Procedia Comput. Sci. 17, 54–61.

Chen, Y., Wu, C., Xie, M., Guo, X., 2011. Solving the sparsity problem in recommender systems using association retrieval. J. Comput. 6, 1896–1902.

Davoodi, E., Kianmehr, K., Afsharchi, M., 2013. A semantic social network-based expert recommender system. Appl. Intell. 39, 1–13.

Devooght, R., Bersini, H., 2016. Collaborative filtering with recurrent neural networks. CoRR abs/1608.07400.

Ding, J., Chen, Y., Li, X., Liu, G., Shen, A., Meng, X., 2016. Unsupervised expert finding in social network for personalized recommendation. In: International Conference on Web-Age Information Management. Springer, pp. 257–271.

Elalfy, D., Gad, W., Ismail, R., 2018. A hybrid model to predict best answers in question answering communities. Egypt. Inform. J. 19, 21–31.

Engin, G., Aksoyer, B., Avdagic, M., Bozanlı, D., Hanay, U., Maden, D., Ertek, G., 2014. Rule-based expert systems for supporting university students. Procedia Comput. Sci. 31, 22–31, 2nd International Conference on Information Technology and Quantitative Management, ITQM 2014.
Faisal, M.S., Daud, A., Akram, A.U., Abbasi, R.A., Aljohani, N.R., Mehmood, I., 2018. Expert ranking techniques for online rated forums. Comput. Hum. Behav..
Ferdous, S.N., Ali, M.M., 2017. A semantic content based recommendation system for cross-lingual news. In: Imaging, Vision & Pattern Recognition, icIVPR, 2017 IEEE International Conference on. IEEE, pp. 1–6.
Fortunato, S., 2010. Community detection in graphs. Phys. Rep. 486, 75–174.
Gajzler, M., 2017. Neural networks in the advisory system for repairs of industrial concrete floors. Computer Assist. Methods Eng. Sci. 18, 255–263.
Ge, M., Elahi, M., Fernaández-Tobías, I., Ricci, F., Massimo, D., 2015. Using tags and latent factors in a food recommender system. In: Proceedings of the 5th International Conference on Digital Health 2015, DH '15. ACM, New York, NY, USA, pp. 105–112.
Georgiou, O., Tsapatsoulis, N., 2010. Improving the Scalability of Recommender Systems by Clustering Using Genetic Algorithms. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 442–449.
Gong, Y., Zhang, Q., 2016. Hashtag recommendation using attention-based convolutional neural network. In: IJCAI, pp. 2782–2788.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.
Grosan, C., Abraham, A., 2011. Rule-based expert systems. In: Intelligent Systems: A Modern Approach. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 149–185.
Gross, J., 2012. 3 - using Facebook, LinkedIn and Twitter for your career. In: Gross, J. (Ed.), Building Your Library Career with Web 2.0. In: Chandos Information Professional Series, Chandos Publishing, pp. 33–53.
Grover, A., Leskovec, J., 2016. Node2Vec: Scalable feature learning for networks. In: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16. ACM, New York, NY, USA, pp. 855–864.
Gubanov, D., Korgin, N., Novikov, D., Raikov, A., 2014. E-expertise: Modern Collective Intelligence. Springer.
Gunawardana, A., Shani, G., 2015. Evaluating recommender systems. In: Recommender Systems Handbook. Springer, pp. 265–308.
Guo, G., 2012. Resolving data sparsity and cold start in recommender systems. In: Masthoff, J., Mobasher, B., Desmarais, M.C., Nkambou, R. (Eds.), User Modeling, Adaptation, and Personalization. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 361–364.
Guo, G., Qiu, H., Tan, Z., Liu, Y., Ma, J., Wang, X., 2017. Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems. Knowl.-Based Syst. 138, 202–207.
Gupta, M.K., Gupta, S., Rawal, R.K., 2016. Chapter 8 - impact of artificial neural networks in qsar and computational modeling. In: Puri, M., Pathak, Y., Sutariya, V.K., Tipparaju, S., Moreno, W. (Eds.), Artificial Neural Network for Drug Design, Delivery and Disposition. Academic Press, Boston, pp. 153–179. http://dx.doi.org/10.1016/B978-0-12-801559-9.00008-9, http://www.sciencedirect.com/science/article/pii/B9780128015599000089.
Gupta, V., Lehal, G.S., et al., 2009. A survey of text mining techniques and applications. J. Emerg. Technol. Web Intell. 1, 60–76.
Harvey, M., Crestani, F., Carman, M.J., 2013. Building user profiles from topic models for personalised search. In: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management. ACM, pp. 2309–2314.
Huang, Z., Zeng, D., Chen, H., 2004. A link analysis approach to recommendation under sparse data. In: AMCIS 2004 Proceedings, p. 239.
Ibrahim, A.M., El-Amary, N.H., 2018. Particle swarm optimization trained recurrent neural network for voltage instability prediction. J. Electr. Syst. Inf. Technol. 5, 216–228. http://dx.doi.org/10.1016/j.jesit.2017.05.001, http://www.sciencedirect.com/science/article/pii/S231471721730020X.
Isinkaye, F., Folajimi, Y., Ojokoh, B., 2015. Recommendation systems: Principles, methods and evaluation. Egypt. Inform. J. 16, 261–273.
Jivani, A., 2011. A comparative study of stemming algorithms. Int. J. Comput. Appl. Technol. 2, 1930–1938.
Johnson, G.H., 1986. The Relative Use of Formal and Informal Information in the Evaluation of Individual Performance (Ph.D. thesis). Texas Tech University.
Kagita, V.R., Pujari, A.K., Padmanabhan, V., Sahu, S.K., Kumar, V., 2017. Conformal recommender system. Inform. Sci. 405, 157–174.
Kalofolias, V., Bresson, X., Bronstein, M., Vandergheynst, P., 2014. Matrix completion on graphs, arXiv preprint arXiv:1408.1717.
Kannan, S., Gurusamy, V., 2014. Preprocessing Techniques for Text Mining.
Karatzoglou, A., Hidasi, B., 2017. Deep learning for recommender systems. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys '17. ACM, New York, NY, USA, pp. 396–397.
Kardan, A., Omidvar, A., Farahmandnia, F., 2011. Expert finding on social network with link analysis approach. In: Electrical Engineering, ICEE, 2011 19th Iranian Conference on. IEEE, pp. 1–6.
Kassim, J.M., Abdullah, R., 2012. Advisory system architecture in agricultural environment to support decision making process. In: 2012 Second International Conference on Digital Information and Communication Technology and it's Applications, DICTAP. IEEE, pp. 453–456.
Kim, H.H., 2013. A Semantically Enhanced Tag-Based Music Recommendation Using Emotion Ontology. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 119–128.
Koch, G., Zemel, R., Salakhutdinov, R., 2015. Siamese neural networks for one-shot image recognition. In: ICML Deep Learning Workshop, vol. 2.
Lamond, D., Crow, R., Chase, J., Doggen, K., Swinkels, M., 1996. Information sources used in decision making: considerations for simulation development. Int. J. Nurs. Stud. 33, 47–57.
Lee, H., Ahn, Y., Lee, H., Ha, S., Lee, S.-g., 2016. Quote recommendation in dialogue using deep neural network. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, pp. 957–960.
Li, H., Lu, Z., 2016. Deep learning for information retrieval. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '16. ACM, New York, NY, USA, pp. 1203–1206.
Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J., 2017. Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, pp. 1419–1428.
Liang, S., de Rijke, M., 2016. Formal language models for finding groups of experts. Inf. Process. Manage. 52, 529–549.
Liang, R., Wang, J., Zhang, H., 2017. Evaluation of e-commerce websites: An integrated approach under a single-valued trapezoidal neutrosophic environment. Knowl.-Based Syst. 135, 44–59.
Likas, A., Vlassis, N., Verbeek, J.J., 2003. The global k-means clustering algorithm. Pattern Recognit. 36, 451–461.
Lin, S., Hong, W., Wang, D., Li, T., 2017. A survey on expert finding techniques. J. Intell. Inf. Syst. 1–25.
Linden, G., Smith, B., York, J., 2003. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Comput. 7, 76–80.
Liu, D.-R., Chen, Y.-H., Kao, W.-C., Wang, H.-W., 2013. Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. Inf. Process. Manage. 49, 312–329.
Liu, M., Liu, Y., Yang, Q., 2010. Predicting best answerers for new questions in community question answering. In: Chen, L., Tang, C., Yang, J., Gao, Y. (Eds.), Web-Age Information Management: 11th International Conference, WAIM 2010, Jiuzhaigou, China, July 15–17, 2010. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 127–138.
Lopez-Nores, M., Blanco-Fern, Y., Pazos-Arias, J.J., Garcia-Duque, J., Martin-Vicente, M.I., et al., 2011. Enhancing recommender systems with access to electronic health records and groups of interest in social networks. In: Signal-Image Technology and Internet-Based Systems, SITIS, 2011 Seventh International Conference on. IEEE, pp. 105–110.
Lousame, F.P., Sánchez, E., 2009. A Taxonomy of Collaborative-Based Recommender Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 81–117.
Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G., 2015. Recommender system application developments: A survey. Decis. Support Syst. 74, 12–32.
Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to Information Retrieval, volume 39. Cambridge University Press.
Minaee, S., Liu, Z., 2017. Automatic question-answering using a deep similarity neural network. CoRR abs/1708.01713.
Mitra, B., Craswell, N., 2017. Neural text embeddings for information retrieval. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17. ACM, New York, NY, USA, pp. 813–814.
Mogotsi, I., 2010. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze: Introduction to information retrieval. Inf. Retr. 13, 192–195.
Mohan, S., Fiorini, N., Kim, S., Lu, Z., 2018. A fast deep learning model for textual relevance in biomedical information retrieval. CoRR abs/1802.10078.
Moreira, C., Wichert, A., 2013. Finding academic experts on a multisensor approach using shannon's entropy. Expert Syst. Appl. 40, 5740–5754.
Nagarnaik, P., Thomas, A., 2015. Survey on recommendation system methods. In: Electronics and Communication Systems, ICECS, 2015 2nd International Conference on. IEEE, pp. 1496–1501.
Nassif, H., Mohtarami, M., Glass, J., 2016. Learning semantic relatedness in community question answering using neural models. In: Proceedings of the 1st Workshop on Representation Learning for NLP, pp. 137–147.
Neshati, M., Fallahnejad, Z., Beigy, H., 2017. On dynamicity of expert finding in community question answering. Inf. Process. Manage. 53, 1026–1042.
Ng, A.Y., Zheng, A.X., Jordan, M.I., 2001. Stable algorithms for link analysis. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 258–266.
Nguyen, H.T., Wistuba, M., Grabocka, J., Drumond, L.R., Schmidt-Thieme, L., 2017. Personalized deep learning for tag recommendation. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 186–197.
Omran, A.M., Khorshid, M., 2014. An intelligent recommender system for long view of egypt's livestock production. AASRI Procedia 6, 103–110, 2nd AASRI Conference on Computational Intelligence and Bioinformatics.
Pal, A., Farzan, R., Konstan, J.A., Kraut, R.E., 2011. Early Detection of Potential Experts in Question Answering Communities. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 231–242.
Park, D.H., Kim, H.K., Choi, I.Y., Kim, J.K., 2012. A literature review and classification of recommender systems research. Expert Syst. Appl. 39, 10059–10072.
Pei, W., Yang, J., Sun, Z., Zhang, J., Bozzon, A., Tax, D.M., 2017. Interacting attention-gated recurrent networks for recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, pp. 1459–1468.

Perozzi, B., Al-Rfou, R., Skiena, S., 2014a. DeepWalk: Online learning of social representations. CoRR abs/1403.6652.

Perozzi, B., Al-Rfou, R., Skiena, S., 2014b. DeepWalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14. ACM, New York, NY, USA, pp. 701–710.

Poria, S., Cambria, E., Gelbukh, A., 2016. Aspect extraction for opinion mining with a deep convolutional neural network. Knowl.-Based Syst. 108, 42–49.

Powers, D.M., 2011. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. Bioinfo Publications.

Pujahari, A., Padmanabhan, V., 2015. Group recommender systems: Combining user-user and item-item collaborative filtering techniques. In: Information Technology, ICIT, 2015 International Conference on. IEEE, pp. 148–152.

Ramezani, M., Yaghmaee, F., 2016. A novel video recommendation system based on efficient retrieval of human actions. Physica A 457, 607–623.

Reihanian, A., Feizi-Derakhshi, M.-R., Aghdasi, H.S., 2017. Community detection in social networks with node attributes based on multi-objective biogeography based optimization. Eng. Appl. Artif. Intell. 62, 51–67.

Rostami, P., Neshati, M., 2019. T-shaped grouping: Expert finding models to agile software teams retrieval. Expert Syst. Appl. 118, 231–245.

Rougas, S., Ashrafizadeh, E., 2018. Search Engine for Journalists to Find Experts. https://expertisefinder.com/.

Sahney, R., Sharma, M., 2018. Electronic health records: A general overview. Curr. Med. Res. Pract. 8, 67–70. http://dx.doi.org/10.1016/j.cmrp.2018.03.004, http://www.sciencedirect.com/science/article/pii/S2352081717302003.

Said, A., Tikk, D., Hotho, A., 2012. The Challenge of Recommender Systems Challenges. ACM, New York, NY, USA, pp. 9–10.

Scherer, D., Müller, A., Behnke, S., 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In: Artificial Neural Networks–ICANN 2010. Springer, pp. 92–101.

Seo, S., Huang, J., Yang, H., Liu, Y., 2017. Representation learning of users and items for review rating prediction using attention-based convolutional neural network. In: 3rd International Workshop on Machine Learning Methods for Recommender Systems, MLRec, SDM'17.

Serdyukov, P., Rode, H., Hiemstra, D., 2008. Modeling multi-step relevance propagation for expert finding. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08. ACM, New York, NY, USA, pp. 1133–1142.

Shani, G., Gunawardana, A., 2011. Evaluating Recommendation Systems. Springer US, Boston, MA, pp. 257–297.

Singh, H., Singh, R., Malhotra, A., Kaur, M., 2013. Developing a biomedical expert finding system using medical subject headings. Healthc. Inform. Res. 19, 243–249.

Steichen, B., Ashman, H., Wade, V., 2012. A comparative survey of personalised information retrieval and adaptive hypermedia techniques. Inf. Process. Manage. 48, 698–724.

Ströele, V., ao, G.Z., Souza, J.M., 2013. Group and link analysis of multi-relational scientific social networks. J. Syst. Softw. 86, 1819–1830.

Su, J.-H., Chang, W.-Y., Tseng, V.S., 2013. Personalized music recommendation by mining social media tags. Procedia Comput. Sci. 22, 303–312, 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013.

Sun, Z., Han, L., Huang, W., Wang, X., Zeng, X., Wang, M., Yan, H., 2015. Recommender systems based on social networks. J. Syst. Softw. 99, 109–119.

Surya Narayana, G., Vasumathi, D., 2017. An attributes similarity-based k-medoids clustering technique in data mining. Arab. J. Sci. Eng..

Tang, J., Hu, X., Liu, H., 2013. Social recommendation: a review. Soc. Netw. Anal. Min. 3, 1113–1133.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q., 2015. LINE: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, WWW '15. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, pp. 1067–1077.

The George Washington University, 2018. GW Expert Finder. https://expert.gwu.edu.

Tsuji, K., Takizawa, N., Sato, S., Ikeuchi, U., Ikeuchi, A., Yoshikane, F., Itsumura, H., 2014. Book recommendation based on library loan records and bibliographic information. Procedia - Soc. Behav. Sci. 147, 478–486, 3rd International Conference on Integrated Information (IC-ININFO).

Tung, Y.-H., Tseng, S.-S., Weng, J.-F., Lee, T.-P., Liao, A.Y., Tsai, W.-N., 2010. A rule-based CBR approach for expert finding and problem diagnosis. Expert Syst. Appl. 37, 2427–2438.

Uysal, A.K., Gunal, S., 2014. The impact of preprocessing on text classification. Inf. Process. Manage. 50, 104–112.

Žalik, K.R., 2008. An efficient k'-means clustering algorithm. Pattern Recognit. Lett. 29, 1385–1391.

Valcarce, D., 2015. Exploring statistical language models for recommender systems. In: Proceedings of the 9th ACM Conference on Recommender Systems. ACM, pp. 375–378.

Valcarce, D., Parapar, J., Barreiro, Á., 2016. Language models for collaborative filtering neighbourhoods. In: European Conference on Information Retrieval. Springer, pp. 614–625.

Véras, D., Prota, T., Bispo, A., Prudêncio, R., Ferraz, C., 2015. A literature review of recommender systems in the television domain. Expert Syst. Appl. 42, 9046–9076.

Wang, X., Huang, C., Yao, L., Benatallah, B., Dong, M., 2018. A survey on expert recommendation in community question answering. J. Comput. Sci. Tech. 33, 625–653.

Wang, Y., Huang, M., Zhao, L., et al., 2016. Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 606–615.

Wang, G.A., Jiao, J., Abrahams, A.S., Fan, W., Zhang, Z., 2013. ExpertRank: A topic-aware expert finding algorithm for online knowledge communities. Decis. Support Syst. 54, 1442–1451.

Wang, J., Sun, J., Lin, H., Dong, H., Zhang, S., 2017a. Convolutional neural networks for expert recommendation in community question answering. Sci. China Inf. Sci. 60, 110102.

Wang, M.-X., Wang, J.-q., 2018. New online recommendation approach based on unbalanced linguistic label with integrated cloud. Kybernetes.

Wang, M.-X., Wang, J.-Q., Li, L., 2017b. New online personalized recommendation approach based on the perceived value of consumer characteristics. J. Intell. Fuzzy Systems 33, 1953–1968.

Wang, X., Yu, L., Ren, K., Tao, G., Zhang, W., Yu, Y., Wang, J., 2017c. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 2051–2059.

Wei, C.-P., Lin, W.-B., Chen, H.-C., An, W.-Y., Yeh, W.-C., 2015. Finding experts in online forums for enhancing knowledge sharing and accessibility. Comput. Hum. Behav. 51, 325–335.

Wu, W., He, L., Yang, J., 2012. Evaluating recommender systems. In: Digital Information Management, ICDIM, 2012 Seventh International Conference on. IEEE, pp. 56–61.

Xie, X., Li, Y., Zhang, Z., Pan, H., Han, S., 2016. A Topic-Specific Contextual Expert Finding Method in Social Network. Springer International Publishing, Cham, pp. 292–303.

Yang, K.-H., Lin, Y.-L., Chuang, C.-T., 2014. Using google distance for query expansion in expert finding. In: Digital Information Management, ICDIM, 2014 Ninth International Conference on. IEEE, pp. 104–109.

Yang, L., Zhang, W., 2010. A study of the dependencies in expert finding. In: Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on. IEEE, pp. 355–358.

Yimam-Seid, D., Kobsa, A., 2003. Expert-finding systems for organizations: Problem and domain analysis and the DEMOIR approach. J. Org. Comput. Electron. Commer. 13, 1–24.

Zhang, J., Tang, J., Li, J., 2007. Expert Finding in a Social Network. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1066–1069.

Zhang, Q., Wang, J., Huang, H., Huang, X., Gong, Y., 2017a. Hashtag recommendation for multimodal microblog using co-attention network. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. AAAI Press, pp. 3420–3426.

Zhang, S., Yao, L., Sun, A., 2017. Deep learning based recommender system: A survey and new perspectives, arXiv preprint arXiv:1707.07435.

Zhang, E., Zhang, Y., 2009. F-measure. In: Encyclopedia of Database Systems. Springer US, Boston, MA, p. 1147–1147.

Zhang, C., Zhang, H., Wang, J., 2018. Personalized restaurant recommendation method combining group correlations and customer preferences. Inform. Sci. 454, 128–143.

Zhao, Z., Wu, Y., 2016. Attention-based convolutional neural networks for sentence classification. In: INTERSPEECH, pp. 705–709.

Zhao, Z., Wu, Y., 2016a. Expert Lookup. https://www.elsevier.com/solutions/expert-lookup.

Zhao, Z., Yang, Q., Cai, D., He, X., Zhuang, Y., 2016. Expert finding for community-based question answering via ranking metric network learning. In: IJCAI, pp. 3000–3006.

Zhao, Z., Zhang, L., He, X., Ng, W., 2015. Expert finding for question answering via graph regularized matrix completion. IEEE Trans. Knowl. Data Eng. 27, 993–1004.

Zhao, D., Zhang, L., Zhao, W., 2016b. Genre-based link prediction in bipartite graph for music recommendation. Procedia Comput. Sci. 91, 959–965, Promoting Business Analytics and Quantitative Management of Technology: 4th International Conference on Information Technology and Quantitative Management (ITQM 2016).

Zhen, L., Song, H.-T., He, J.-T., 2012. Recommender systems for personal knowledge management in collaborative environments. Expert Syst. Appl. 39, 12536–12542.

Zheng, N., Li, Q., 2011. A recommender system based on tag and time information for social tagging systems. Expert Syst. Appl. 38, 4575–4587.

Zheng, Z., Lu, D., Chen, Q., Yang, H., Xiang, Y., Pan, Y., Zhong, W., 2018. Answer selection in community question answering by normalizing support answers. In: Huang, X., Jiang, J., Zhao, D., Feng, Y., Hong, Y. (Eds.), Natural Language Processing and Chinese Computing. Springer International Publishing, Cham, pp. 672–682.

Zheng, C., Zhai, S., Zhang, Z., 2017. A deep learning approach for expert identification in question answering communities. CoRR abs/1711.05350.

Zhou, X., Hu, B., Chen, Q., Wang, X., 2018. Recurrent convolutional neural network for answer selection in community question answering. Neurocomputing 274, 8–18.

Zhou, G., Zhao, J., He, T., Wu, W., 2014. An empirical study of topic-sensitive probabilistic model for expert finding in question answer communities. Knowl.-Based Syst. 66, 136–145.