Chaos, Solitons and Fractals 151 (2021) 111260

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

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

BERTERS: Multimodal representation learning for expert recommendation system with transformers and graph embeddings

N. Nikzad-Khasmakhi[a], M.A. Balafar[a,*], M. Reza Feizi-Derakhshi[a], Cina Motamed[b]

[a] Department of Computer Engineering, University of Tabriz, Tabriz, Iran
[b] Department of Computer Science, University of Orleans, Orléans, France

ARTICLE INFO

Article history:
Received 17 April 2021
Revised 21 June 2021
Accepted 6 July 2021
Available online 9 August 2021

Keywords:
Multimodal representation learning
Expert recommendation system
Transformer
Graph embedding

ABSTRACT

An expert recommendation system suggests relevant experts of a particular topic based on three different scores authority, text similarity, and reputation. Most of the previous studies individually compute these scores and join them with a linear combination strategy. While, in this paper, we introduce a transfer learning-based and multimodal approach, called BERTERS, that presents each expert candidate by a single vector representation that includes these scores in itself. BERTERS determines a representation for each candidate that presents the candidate's level of knowledge, popularity and influence, and history. BERTERS directly uses both transformers and the graph embedding techniques to convert the content published by candidates and collaborative relationships between them into low-dimensional vectors which show the candidates' text similarity and authority scores. Also, to enhance the accuracy of recommendation, BERTERS takes into account additional features as reputation score. We conduct extensive experiments over the multi-label classification, recommendation, and visualization tasks. Also, we assess its performance on four different classifiers, diverse train ratios, and various embedding sizes. In the classification task, BERTERS strengthens the performance on Micro-F1 and Macro-F1 metrics by 23.40% and 34.45% compared with single-modality based methods. Furthermore, BERTERS achieves a gain of 9.12% in comparison with the baselines. Also, the results prove the capability of BERTERS to extend into a variety of domains such as academic and CQA to find experts. Since our proposed expert embeddings contain rich semantic and syntactic information of the candidate, BERTERS resulted in significantly improved performance over the baselines in all tasks.

1. Introduction

Recently, the shadow of recommendation system (RS) has appeared on various domains and applications. On the other hand, significant new advances in deep learning approaches [1,2] have important effects on the tremendous success of the recommendation system [3,60]. The overall structure of a RS follows a set of phases including collection, learning, and recommendation [4,5]. In the first phase, appropriate resources that comprise the relevant information of users are selected. Then, a learner (supervised or unsupervised learning) analyzes the users preferences and extracts their behavioral patterns. The final phase recommends the items or entities that are the most similar to the users' interests. It is important to recognize that, within a common core structure of RS, there are variations from application to application. Some of the most sophisticated and heavily used RSs in the industry are Last.fm, YouTube, and Amazon.

Furthermore, we can find the footprint of RS in the knowledge management system where RS tries to specify experts who have the most relevant knowledge about a particular topic [6,7]. This category of RS is called expert recommendation system (ERS) or expert finding system. So, it is obvious that an ERS has similar phases compared to general RSs. Figure 1 demonstrates the basic elements of an expert recommendation system. An ERS takes a user topic or query, traces a set of candidates' expertise, learns their expertise patterns, and finally produces a list of experts sorted by a score. As it is seen in the figure, the candidates' expertise is defined as content-based and non-content-based information [8]. Content-based information is candidates' shared textual content like their articles, questions, answers, and so on. In contrast, candidates' interactions with each other in social networks make non-content-based information. The functionality of the learner element in the expert recommendation system is

* 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), motamed@free.fr (C. Motamed).

https://doi.org/10.1016/j.chaos.2021.111260
0960-0779/

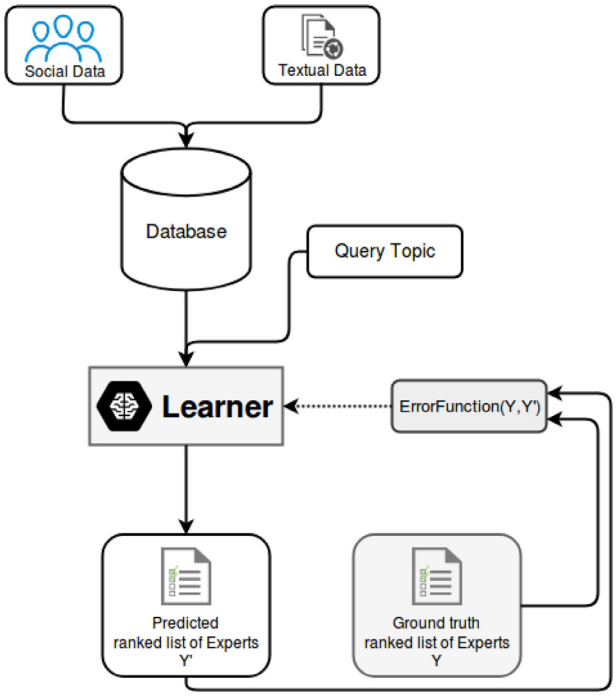


Fig. 1. The phases of an expert recommendation system [7].

analyzing the content-based and non-content-based information and mapping the experts expertise to his/her corresponding score. The learner can be a supervised, unsupervised or semi-supervised method, but almost all studies in ERS propose or use a supervised learner which models relationships and dependencies between the candidates' expertise and their scores, then the ERS can predict scores for new candidates based on those relationships which it learned from the previous candidates.

Each candidate's score indicates the degree of this candidate's relevant expertise with the given topic and consists of three different scores authority, text similarity, and reputation [7]. Authority score measures the influence and popularity of users in social networks. On the other hand, text similarity represents the level of the user's knowledge according to contents published by the user. Moreover, reputation score is obtained from the user's historical activity in social networks. It implies that candidates with high reputation scores share more knowledge and information with others in the communities. The final scores of candidates arise from the combination of these scores which determine the experts [7].

Depending on the application scenarios, each ERS has its own target, contextual information, set of phases and scores. For example, the attempt in an academic environment is to detect researchers who have the subject areas related to the query. This detection can be based on the content of the articles published that is defined as text similarity score and their co-author relations in different papers which is used to compute authority score. However, in Community Question Answering (CQA), the main goal is to find the users with expertise and willingness to answer the given questions in terms of the content of the question asked and the answer posted by them and their question-answer relations [9]. The content of questions and answers is explored to calculate the text similarity. Moreover, the authority and reputation scores are derived from the useruser relationship graph and the users historical question-answering records such as the number of questions and answers the user has posted, respectively [8,10].

With a brief look at previous studies, it can be concluded that there are three different outlooks on ERSs. In one of the attitudes, studies have focused on the textual expertise of candidates. These works have used text mining or information retrieval techniques and selected those as experts whose published items are semantically relevant to the query [11]. That means these approaches just consider text similarity and ignore the reputations of candidates and their authority values to predict the users' performance. Hence, most of document-based methods take into account the problem of finding experts from the viewpoint of NLP task and explore text representation models such as language models, document models, document embedding and so on to learn users' expertise representations.

On the other hand, some other researchers have investigated the social relations between candidates and represented their connections as a graph [12]. After that, graph analysis techniques such as page ranking algorithms or graph embedding approaches are applied to this graph to identify important candidates. In other terms, these approaches emphasize on the authority score and disregard the text similarity and reputation scores [13]. As a result, these approaches cannot recommend the good candidates. For example, if two candidates publish articles related to the same field, these two authors are not considered to be similar by these approaches [14].

Moreover, recent studies have shown that the combination of different types of expertise information has notable performance compared to others. A number of these studies have integrated textual expertise and social network connection information to infer the text similarity and authority scores. Also, to achieve higher accuracy, a few investigations in the CQA domain have suggested the usage of the heterogeneous network which is a combination of the users' interactions in social networks and their question answer relationships in CQA besides bearing in mind the content of questions and answers. In most cases in the final step a combination strategy combine these different features into a single expert ranking score. A large number of approaches use a weighted linear combination method for this purpose. A weight is assigned to each score based on its importance that may be different based on the application scenario. On the other hand, some other hybrid models create multiple objective functions for text similarity and authority scores and then merge them and pass it to the training to learn a single objective function. That means that these approaches do two different tasks and share some part of the model between two tasks [15]. So, the model trains a classifier that employs the gradient descent optimizer. In this case, the approach optimizes some linear combination of two losses that linear combination weight would be a hyper parameter and can be tuned. This comes from the nature of gradient that is a linear operation. Totally, training two models for different loss trade-offs is very inefficient because of its requirements for keeping around two models in the training process [16]. Hence, the important point in all hybrid models is that there is no evidence and reason that the relation between authority, text similarity and reputation scores should be linear.

Although it is necessary to take the text content and user relationship into consideration simultaneously, the way of merging these modalities to a single score or representation is also significant. To address this issue, in this article, we provide a transfer learning-based and multimodal approach that presents each expert candidate by a single vector representation that includes the authority, text similarity, and reputation scores in itself. In other words, instead of separately calculating scores and merge them to create a final score for each candidate, our proposed approach learns a representation for each candidate that presents altogether the candidate's level of knowledge, popularity and influence, and history.

In this research, we aim to find academic experts that whether using a multimodal learning approach provides an effective solution for ERS or not. Also, the other purpose of our work is to solve the expert finding problem as a multi-label classification task. In

such a way, we combine text (articles) and graph (co-author connections) information in a multimodal approach. The text component focuses on the text similarity score and determines the level of expertise of candidates. To convert this textual information into vector, we take the advantages of Transformers including BERT [17], Sentence-BERT (SBERT) [18], and Universal Sentence Encoder [19] Transformers. On the other hand, the co-author graph is a good solution to find candidates who share more knowledge with others in the communities and are more active. To learn the structural vector from the graph and capture the authority score, three graph embedding techniques including ExEm [20], DeepWalk [21] and Node2vec [22] are used. Also, the candidate's normalized h-index value is added as extra feature and reputation score. Then, the captured fusion features are fed into a classifier to learn the expert embeddings. We examine four different classifiers Fully connected, Random Forest [23], Support Vector Machine (SVM) [24], and Logistic Regression for the classification part. Finally, we determine the effectiveness of BERTERS on the multi-label classification, recommendation and visualization tasks. The major contributions of this paper can be summarized as follows:

- To the best of our knowledge, we provide the first multimodal approach that presents each expert candidate by a single vector representation. This single representation indicates the authority, text similarity, and reputation scores. In other words, rather than individually defining three scores and combining them to a single score, BERTERS determines a representation for each candidate that presents the candidate's level of knowledge, popularity and influence, and history.
- To the extent of our knowledge, this is the primary prospective study that directly uses both transformers and the graph embedding techniques to convert the content and non-content information into low-dimensional vectors. These vector representations illustrate the candidates' text similarity and authority features. Also, BERTERS attaches the h-index value as reputation score to obtain the final representation.
- The usefulness of expert representations comes when the goal is to compare expert candidates. Expert vectors represent experts as multidimensional continuous floating point numbers where experts with similar expertise have similar embeddings and are mapped to proximate points in geometric space.
- Proposed expert embeddings can benefit a lot of applications such as expert classification, expert clustering, expert recommendation, detecting communities of experts, link prediction that predicts whether two experts will corporate with each other as co-author in the future.
- Moreover, in this research we observe the problem of finding experts in the form a classification task.
- Our proposed model can be extended to different environments such as academic and CQAs to find experts.

The rest of the paper is structured as follows: Section 2 reviews the related works. Section 3 discusses the background of the research. Section 4 presents our proposed method and explains it in detail. The descriptions of the dataset and the tasks that are used to test our proposed method and parameter setting are presented in Section 5. Section 6 provides experimental results. Section 7 indicates the further comprehensive discussion about results, key findings of our work and future. Finally, Section 8 concludes the paper.

2. Related work

In this section, we review the approaches proposed for ERS. We group these models into three categories, based on their main outlooks: document-based, graph-based and hybrid models. The subsections bellow will explain the underlying methodology and existing approaches for the specified categories. We summarize the studies in Table 7. We also add some extra information extracted such as citation and year from Google Scholar. Moreover, in case of reader curiosity, we highly recommend reading [7,25,26] that explain in more detail and are dedicated to review all the related researches in this scope.

2.1. Document-based models

Document-based models are intended to compare the characteristics of the content contained in the published items associated with a candidate and the query. Document-based models work well where capturing the level of experts' knowledge in the field of the topic query is the goal. Briefly, these approaches focus on the text similarity score without considering the popularity and history of candidates. Document-based methods present the problem of finding experts from the viewpoint of NLP task and learn the semantic representation of candidates' published content using models such as language models, document models, document embedding. A number of works employed traditional document representation methods such as TF-IDF, Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA) to determine the text similarity scores of candidates using vector similarity metrics like cosine similarity.

In study [27], authors suggested a framework to automatically direct new questions to the best experts based on tracking their answering history in the community. For this goal, they employed different methods consisting of language models with Dirichlet smoothing, TF-IDF, LDA, and Segmented Topic Model (STM). In another research [28], authors applied LDA method to collect the topics of documents. After that, the probability of each candidate query is calculated based on the extracted topics for each query. Experts are sorted according to this probability. Authors in paper [29] emphasized the dynamic aspects of the expert finding. The authors considered four content features including topic similarity, emerging topics, user behavior, and topic transition features to predict the best ranking of experts in the future. Study [30] presented a tag-LDA approach to model the candidate topic distribution. Additionally, authors in [31] suggested a novel approach LDA to determine the main underlying topics of each political speech, and to distribute the related terms among the different topic-based subprofiles.

Also, there are other interesting document-based models that explored Word2Vec and Doc2vec two well-known methods for producing word and document embedding models. Research [32] found the experts in the Faculty of Computer Science in Universitas Indonesia by representing the query and expertise of the lecturers by the combination of Word2Vec and Doc2vec. Some other studies combined Word2Vec with deep learning approaches to find experts in CQA scope, especially. Wang et al. [33] proposed a model to find experts in CQA by using Word2Vec to represent the question and user profiles. Then, authors applied convolutional neural networks to predict which users were more likely to give the best answer for the newly posted question. Authors in [34] created a profile for each candidate expert based on his activities by using a long short-term memory(LSTM) neural network. Then, experts' shape of expertise is determined by learning the pattern of changes in their expertise trees.

Despite the fact that document-based approaches are helpful in finding knowledgeable candidates, they cannot detect the important or influential experts in the social networks. In simple words, these methods consider the expert recommendation task as a content-based expert finding problem and disregard the authority and reputation scores. Hence, they cannot provide high-quality experts. On the other hand, non of these methods use the efficiency of transformers in their strategies. Transformers produce

Table 1
Dataset information.

Dataset	$\|V\|$	$\|E\|$	Labels	# Articles	# Authors with articles	# Questions	# Answers	# Users
Scopus	27,473	285,231	27	472,566	9,378	-	-	-
Quora	-	-	-	-	-	444,138	887,771	95, 915

better contextual word representation and are fast to train and easy to parallelize.

2.2. Graph-based models

Document-based models recognize expertise patterns across documents, whereas graph-based approaches learn to recognize patterns across graphs. Graph-based models work well where authority scores of candidates are important. Authority score measures the influence and popularity of candidates in social networks [7]. The graph-based methods formulate the problem of ERS from the perspective of a graph $G(V, E)$, where V denotes a set of candidates and E is a set of edges among the nodes that comes from the interactions between candidates. Depending on the applications at hand, nodes can represent candidate experts of various types such as academic candidates or the best answerers. On the other hand, edges represent different kinds of relations between the candidates, like question posters and repliers relations in CQA, follower-following connections in social networks, co-author relationships, etc.

Most previous graph-based approaches applied link analysis techniques such as PageRank and HITS to measure the similarity between candidates with a topic query, calculate candidates' scores and make recommendations. Fu et al. [35] proposed an expertise propagation algorithm to build the relationship between candidates. Their proposed approach is very similar to PageRank. Additionally, there are some other researches that benefit from the graph embedding techniques. Sun et al. [36] proposed a novel asymmetric transitivity preserving directed graph embedding method which was factorization based. Authors applied their model to the task of expert finding to estimate the user expertise and route newly posted questions to users with the suitable expertise and interest in CQAs.

Moreover, there are some other papers focusing on detecting the top-K influential users as candidate experts in communities [37]. For example, in e-commerce websites consumers can comment or review products online. The goal of expert finding in e-commerce applications is to find consumers who are influential and provide high-quality and useful comments or reviews to a new product or service [7]. These expert reviewers are called influential users. Mumtaz and Wang [38] proposed a simple technique to find the influential node set in a network with the largest betweenness centrality. Paper [39] reviewed the existing works on identifying top-k influential and significant nodes.

The graph-based approaches perform authority score ranking and find the influential candidates in the social network. The main drawback of these strategies is that they fail to consider each expert candidate's topical expertise and reputation. Briefly, they do not take into account the level of knowledge of candidates in finding experts, and direct their attention to the relations between candidates.

2.3. Hybrid models

Hybrid models have drawn a lot of attention for ERS in recent years. These methods have been developed to combine features extracted from the documents (or questions and answers), and features obtained from candidates' social network communications to formulate a recommendation. It should be noted that hybrid models need to use a feature-combination method to merge content and non-content expertise and calculate different scores. This section reviews some of the most prominent hybrid models which created new state-of-the-arts on ERS. Zhou et al. [13] considered the candidate's expertise and reputation score to recommend experts. They proposed a user-topic model to analyze the content of the questions and answers. Moreover, the authors introduced a topic-sensitive method to reflect both the link structure and the topic relevance between questioners and answerers. Liu et al. [10] merged knowledge, reputation and authority scores of candidates to produce a recommended expert list. Knowledge score shows the similarity of the profile and the target question. Moreover, the number of answers and the best answers given by candidates are used to find the reputation score. Finally, the authority score is calculated using HITS and Page Rank approaches. Xie et al. [40] used LDA and HITS algorithms to extract topical feature. The suggested method evaluated social behaviour, time and location factors in order to extract contextual features. Finally, a SVM algorithm was used as a scoring function. Furthermore, there are other interesting hybrid models such as CQARank [41], ExpertRank [8], HSNL [42], GRMC [43], RMNL [44], Expert2Vec [11], MMSE [45], ExpFinder [46].

The hybrid models have achieved high accuracy on many ERS benchmarks. But, the important point in these approaches is how to combine content and non-content elements to detect experts. In most previous hybrid models, a linear combination strategy is explored to join different scores obtained from different features. Based on the usage scenarios, each score may have high priority and therefore a higher weight is assigned to that score.

Unlike most previous researches that use a weighted linear combination strategy, our proposed hybrid model, a transfer learning-based and multimodal approach, creates vector representations including all three scores for candidates. These semantic expert embeddings provide an effective and efficient way to solve the problem related to the fusion of features. These vectors allow to compare expert candidates and also find similar experts by measuring similarity from embeddings using similarity measures such as Euclidean distance, Cosine and dot product.

3. Background

In this section, we discuss the concepts which organize the background of our study. In this way, firstly, the text representation methods, BERT, SBERT and USE Transformers, are explained. After that, the graph embedding techniques, ExEm, DeepWalk and Node2vec are introduced.

3.1. Text representation learning

In recent years, researchers have made a lot of efforts in extracting features from text data and have proposed many models including neural embedding, attention mechanism, self-attention, and Transformer. As investigated in many papers, the sequential processing of text and the computational cost of obtaining remarkable relationships between words in a sentence are two issues that RNN and CNN models are encountered with, respectively. On the other hand, transformers eliminate these bottlenecks by assigning in parallel an attention score to each word in a document to

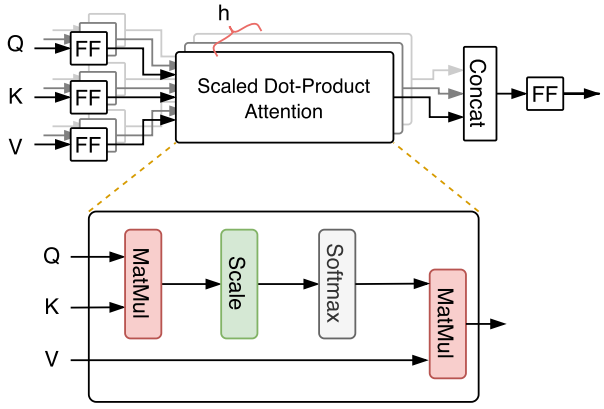


Fig. 2. The Transformer model architecture [49].

consider the impact of words on each other [47]. Figure 2 illustrates the architecture of the transformer model that comprises both encoding and decoding components which are all identical in structure. These components include the stacked layers. For example, the encoding component is a stack of encoders where each stack layer is broken down into two sub-layers. Each sub-layer has a multi-head attention layer and a feed-forward neural network. The multi-head attention layer extracts the dependencies between representation pairs regardless of the distance between them in the sequence and is more effective than single-head attention [47,48,61]. The outputs of the attention layer are injected into the feed-forward. For each set of queries Q, keys K, and values V, the multi-head attention module applies h attention functions which are the scaled dot-product attention as shown in Eq. (1).

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

One of the most widely used transformer models is BERT Transformer [17] that is the new state-of-the-art sentence embedding model [47]. The BERT Transformer architecture is shown in Fig. 3. A masked language modeling task is used for training BERT. It randomly selects some tokens in a text sequence for masking, and then independently retrieves the masked tokens by conditioning on the encoding vectors which are the outputs of a bidirectional Transformer. For using BERT, firstly, two tokens, that are known as [*CLS*] and [*SEP*], are added at the beginning and the end of the text input, respectively. After that, the input flows through the two transformer layers. The output of the last transformer layer is the embedding of the input. Briefly, BERT model has two parameters h and L. h is the size of the output embedding vector and L shows the number of stacked layers in each component.

In addition to BERT transformer, SBERT and USE are two other text embedding techniques used in this study to provide a vector representation of documents. SBERT is the modified version of the pretrained BERT model that makes use of siamese and triplet network structures, as shown in Fig. 4a, to create semantically meaningful sentence embeddings. The siamese network architecture allows SBERT to derive fixed-sized vectors for input sentences. In order to generate the sentence embeddings, the authors fine-tuned SBERT on Natural Language Inference (NLI) data [18]. On the other hand, USE converts sentences into vector representations by two different multilingual modules with the potential of transfer learning to other NLP tasks. Although these two variants of USE can support a wide variety of applications including clustering and text classification, they are designed with different goals. One version, as shown in Fig. 4b, uses transformer architecture that aims to provide high accuracy at the cost of greater model complexity. The second model of USE produces sentence embeddings by a deep averaging network (DAN) that is computationally less expensive [19]. In this study, we employ the universal sentence encoder based on the transformer to obtain the sentence representation.

3.2. Node representation learning

One of the key concepts in the analysis of social networks is the idea of presenting the knowledge inside them as a graph structure [50]. On the other hand, in recent times, one of the most widely used graph analysis approaches is graph embedding. Graph embedding represents the graph nodes as low-dimensional vectors [51,52,58]. It gives us a deeper vision to analyze users' activity patterns and their relationships in social networks [59]. A number of recent techniques have developed to embed graph nodes. In our study, we focus on three embedding techniques including DeepWalk [21], Node2vec [22], and ExEm [20] that employ random walks on a graph to obtain node representations.

DeepWalk is the first effort proposing the deep learning technique into graph analysis. Because the random walks can govern the structure of the graph, DeepWalk uses a stream of short random walks to model the graph. It considers each random walk as a sentence and the graph nodes as words. Therefore, authors can generalize the idea of language modeling in NLP to explore the graph. The aim of language modeling is to compute the probability of a sentence or the sequences of words as shown in Eq. (2).

$$\begin{aligned} P(w) &= P(w_1 w_2 ... w_m) = \prod_{i=1}^{m} P(w_i | w_1 w_2 ... w_{i-1}) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1 w_2)...P(w_m|w_1 ... w_{m-1}) \end{aligned} \tag{2}$$

To transfer the language modeling into the graph, the task is to estimate the probability of Eq. (3).

$$\begin{aligned} P(v_i|(v_1 v_2 ... v_{i-1})) &= P(v_i|\Phi(v_1)\Phi(v_2)...\Phi(v_{i-1})) \\ &= \prod_{i=1}^{m} P(v_i|\Phi(v_1)\Phi(v_2)...\Phi(v_{i-1})) \end{aligned} \tag{3}$$

where Φ is the low-dimensional representation of each node in the graph. So, each vertex v_j is converted to its representation vector $\Phi(v_j)$ by feeding the walks as inputs into the SkipGram neural network to maximize the probability of node's neighbors in the walk [21].

In Node2vec, the authors introduce a flexible strategy to generate the node's neighborhood. They design a biased random walk procedure based on the concept of the breadth-first and depth-first search algorithms. In this model, two parameters p and q help Node2vec control over the search space. While parameter p manages the likelihood of immediately revisiting a node in the walk, indicator q supervises the distances from a given source nodes [20,22].

Also, ExEm is a random walk based technique that hires dominating nodes to modify the random walk strategy used in DeepWalk and Node2vec with regard to the homophily and structural role objectives. ExEm creates a set of random walks that contains at least two dominating nodes. The first dominating node ensures that ExEm selects a node within this dominating node community; so ExEm learns to embed nodes of a community into similar vectors. Moreover, with the help of the second dominating node, ExEm can observe nodes that are far from starting node and belong to the other clusters and this is what the homophily role says. On the other hand, the structural role objective emphasizes that nodes with the same roles should be embedded closer. Since dominating nodes are the heads of their communities and play the same roles, this allows ExEm to perceive the nodes with the same roles in each sampled path and embed them into similar embeddings. There are three variants of the ExEm model including ExEm_{w2v}, ExEm_{ft} and

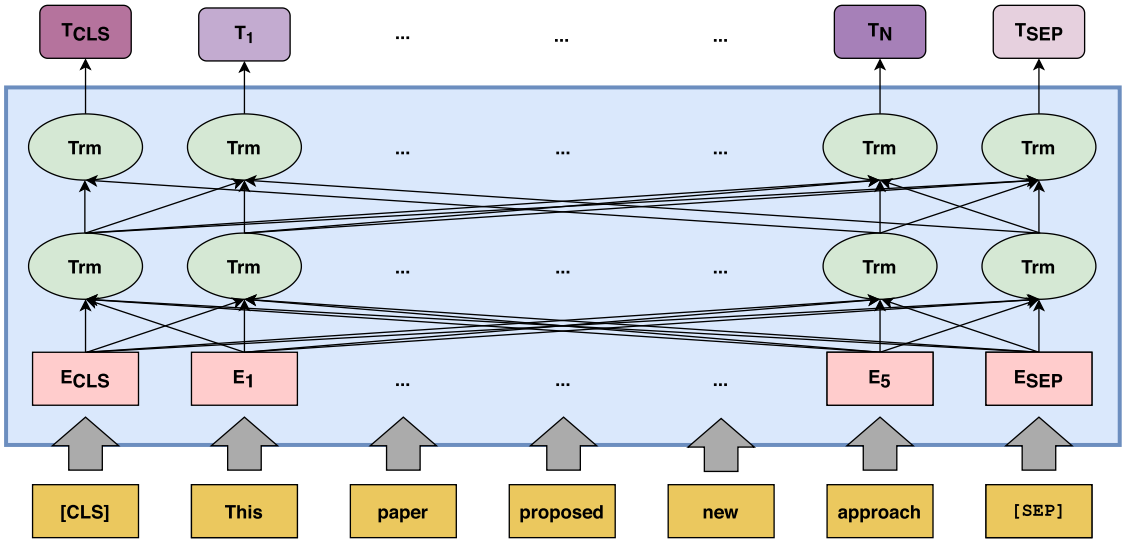
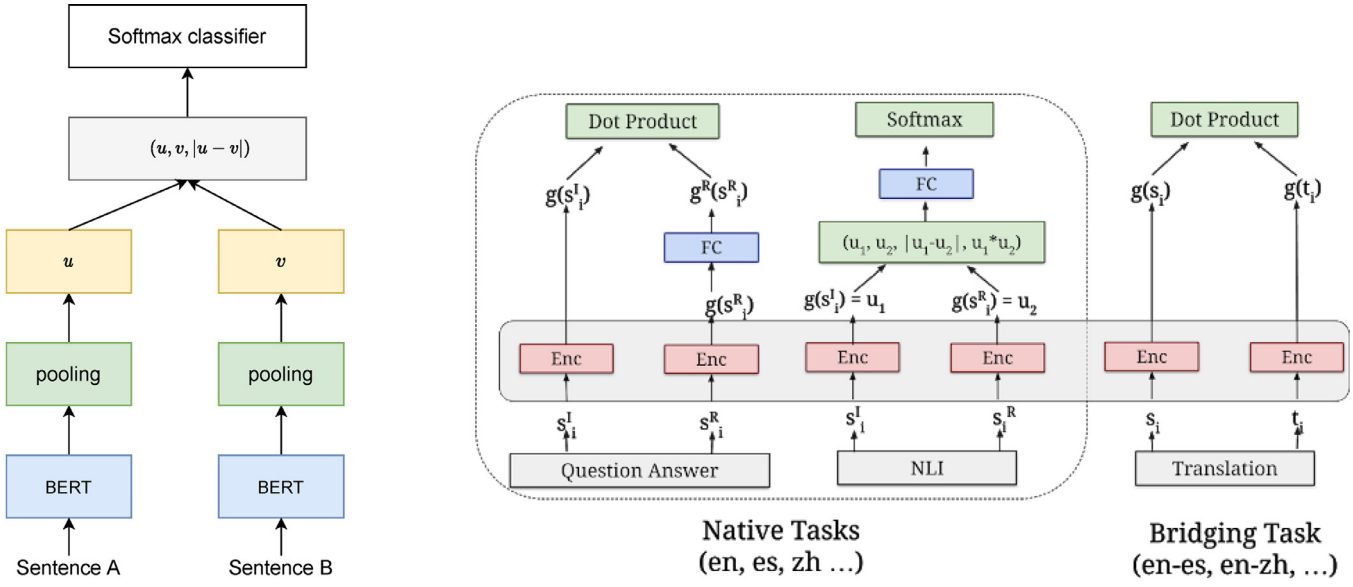


Fig. 3. The BERT Transformer architecture.



(a) The structure of SBERT [18]

(b) The structure of the Universal Sentence Encoder [19]

Fig. 4. SBERT and USE architectures; these architectures present the structure of models for training and inference.

$ExEm_{com}$ that uses Word2Vec, fastText, and their combination to train the Skip-gram neural network, respectively [20].

4. Proposed method

The aim of this paper is to design a new hybrid model, called BERTERS, that is able to find academic experts. BERTERS proposes a transfer learning-based and multimodal approach that presents expert candidates in form of low-dimensional vectors with respect to authority, text similarity and reputation scores. The overall structure of BERTERS is shown in Fig. 5. In the first step, BERTERS extracts the adequate dataset from Scopus which is the largest abstract and citation database. The gathered dataset includes the content and non-content features of expert candidates such as their published articles, subject areas, affiliations, h-index, and their co-author interactions. In the next phase, BERTERS takes as inputs the articles, the co-author connections and h-index that have various types (e.g., text and graph). These different modalities create the required content and non-content features. So, these modalities enable a multimodal deep learning approach to create comprehensive and meaningful representations of expert candidates. To capture candidates' representations from these different modalities, BERTERS is comprised of three different neural networks: one for document representation generation, the other one for node representation generation, and the third one for learning a shared representation between modalities. Each feature is separately obtained from the respective neural network and then merged with other features to create a single representation for each candidate. Then, the task of recommendation is modeled as a classification problem where candidates' subject areas are defined as their labels to learn the candidate embeddings. Finally, BERTERS provides a list of candidates as experts via collaborative filtering.

To the best of our knowledge, BERTERS is the first recommendation model for ERS that employs multimodal learning and transformers. As of another meaning, BERTERS perceives the ERS as a vision of a multi-label classification task using multimodal and

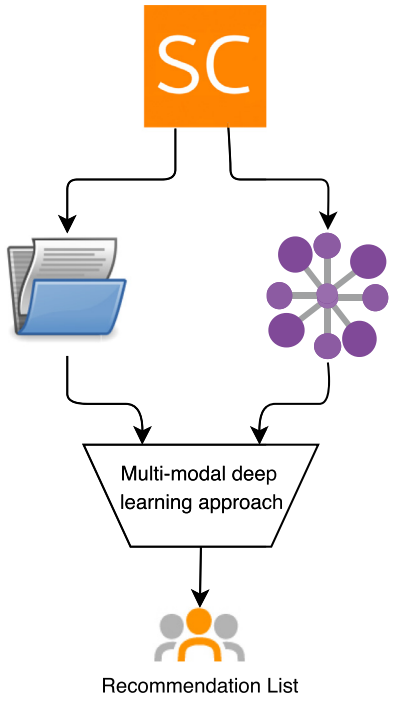


Fig. 5. The overall structure of BERTERS.

transfer learning that employs BERT, SBERT and USE transformers to capture content feature or text similarity score. Also, BERTERS takes advantage of graph embedding technique to learn the non-content features considered as authority score. To enhance the accuracy of recommendation, the candidate's h-index score is taken into account as reputation score. Eventually, all these three features are combined to embed each candidate into a vector space which contains rich semantic and syntactic (structure) information of candidates. After that, BERTERS presents three widely used evaluation tasks such as classification, recommendation and visualization to measure the quality of candidate embeddings. The following subsections describe the procedures of BERTERS in detail.

4.1. Model architecture

As it was mentioned previously, in this study, we introduce a transfer and multimodal learning approach that considers the ERS as a multi-label classification task. Fig. 6 shows how BERTERS treats the expert recommendation task as a classification problem. From this viewpoint, the prediction problem becomes accurately classifying a specific expert candidate where the candidates' subject areas are defined as their labels. This model can be formalized as computing the probability of all possible subject areas for an expert candidate based on the average of all document embeddings D_e, candidate social connection embedding N_e, and h-index Hi:

$$P(C_{i_{sa}}|D_e, N_e, Hi) = P(C_{i_{sa}}|[D_e; N_e; Hi]) \approx P(C_{i_{sa}}|E) \tag{4}$$

where $C_{i_{sa}}$ is a candidate C_i with subject areas of sa, and E is computed based on applying three dense layers with ReLU (Rectified Linear Units) function on the concatenation of D_e, N_e and Hi, as defined in Eq. (5). In other sense, BERTERS learns the expert candidate embeddings E as a function of text similarity, authority and reputation scores.

$$E = ReLU(ReLU(ReLU([D_e; N_e; Hi]W)W)W) \tag{5}$$

where $ReLU$ is a linear activation function that transfers directly the input into the output if it is positive, otherwise, it returns zero. Also, W presents the weights of the network. The direct analog is to estimate the likelihood of subject areas of a candidate based on E. Hence, a Sigmoid classifier applies on the embedding E. Eq. (6) shows this probability.

$$P(C_{i_{sa}}|E) = Sigmoid(E) \tag{6}$$

Moreover, an important point is that ideally a proposed approach for an ERS should take into account the content and non-content features of candidates to calculate three scores. Content feature indicates that how much a candidate's shared textual context is similar to the input topic. Based on this feature, BERTERS learns that all candidates whose published items have similar topics, should have similar expert embeddings. On the other hand, the non-content features guarantee that the candidates with similar social activities should be embedded closely together. In other words, the content feature emphasizes on the knowledge of candidates, whereas the non-content one focuses on the candidate connectivity of the collaboration network. We observe that using a multimodal approach allows BERTERS to convert expert candidates into vectors by considering the above features. Note that one of the major improvements of BERTERS compared to other methods is its extensibility. For example, in case of CQA, BERTERS can be applied with a very low effort. The only requirement is distinguishing the content and non-content features of the task at hand. In the following subsections, we describe how BERTERS presents candidates as low-dimensional vectors with regard to different scores and features in the academic and CQA systems.

4.2. Document representation generation

As it can be concluded from Fig. 5, one of the BERTERS modalities is textual information that comes from the articles published by candidates. It aims at extracting distinguishing textual expertise of candidates. Using text modality as a content feature helps BERTERS to learn that candidates with similar topics of interest should be embedded closer. In other words, text learning ensures the text similarity score and provides more information to assess context similarity between candidates' expertise.

Exploring transfer Learning using transformers is becoming a common approach in NLP. Therefore, BERTERS learns the representation of each document using transfer learning from pre-trained networks of three transformer models including BERT, SBERT and USE demonstrated in Section 3.1. The text information of candidates' expertise can be composed of the article's titles, abstracts and keywords. The mentioned components of an article related to each candidate make the input of each transformer as indicated in Fig. 6. The input article passes through the layers of each transformer, and the output that is a unique vector of each document is created. Consequently, a candidate's text similarity score is represented by a high-dimensional vector D_e which is the average of his/her all article embeddings.

It is worth noting that we can extend this procedure for the ERS in CQA. For this purpose, the questions asked and the answers posted by the candidates are fed into the inputs of the transformers. After that, the average of these embeddings are used as the text modality value.

4.3. Node representation generation

Learning features of modalities is the foundation of multimodal deep learning approaches. As explained before, another modality in BERTERS fetches from the co-author network. The co-author relationships between candidates help BERTERS to consider the influence and popularity of candidates in their learning representations. For this purpose, BERTERS transforms the non-content features extracted from Scopus including candidates' id, their fields of inter-

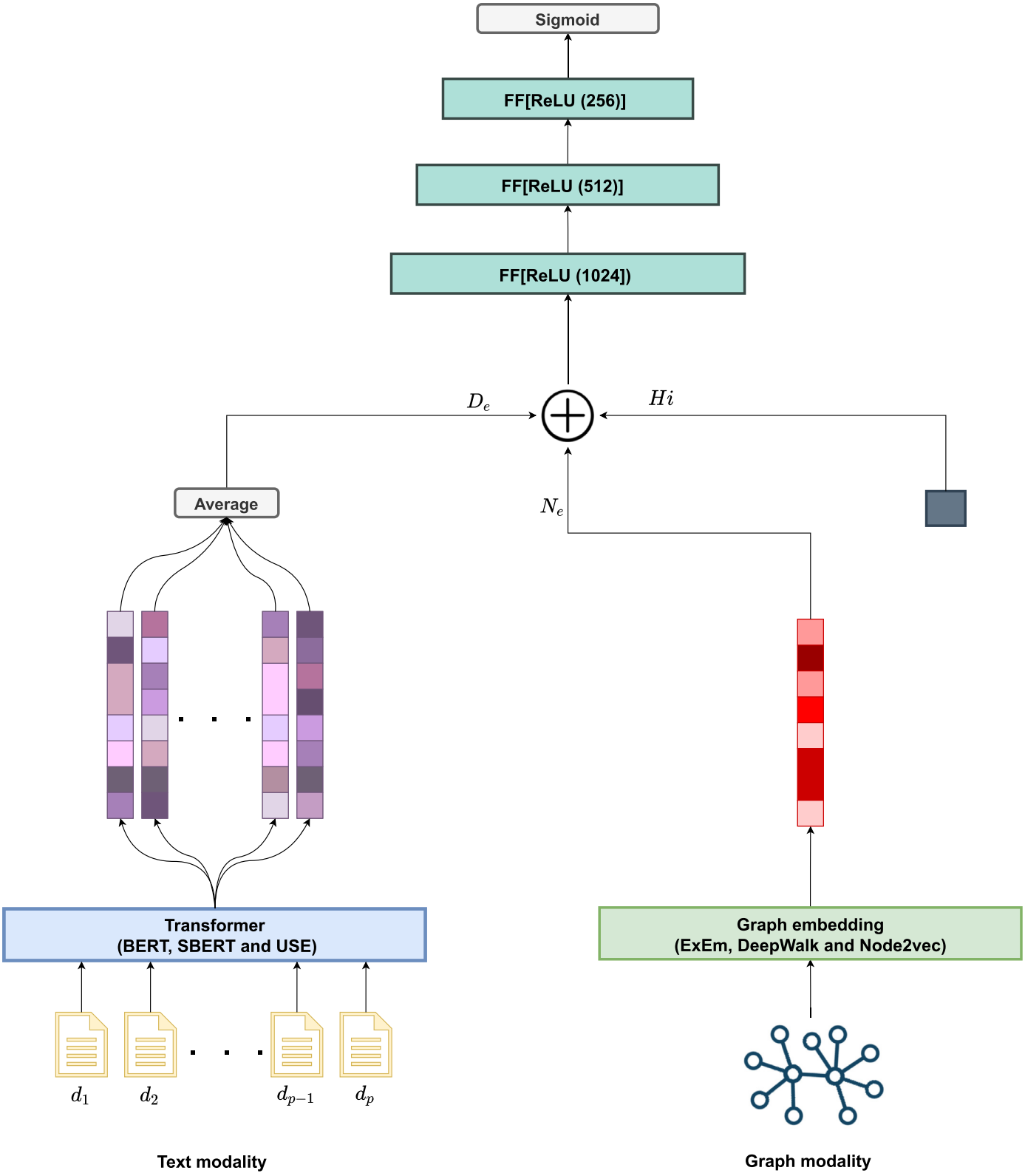


Fig. 6. Multimodal architecture of BERTERS.

est, and their connections into a labeled collaborative graph format. Candidates and their subject areas are defined as nodes and their labels, respectively. The graph edges originate from the authors co-author collaborations. In order to interpret information of the constructed collaborative network and get the node representations of candidates, BERTERS takes the advantage of the graph embedding techniques that are described in Section 3.2. The candidate's node embedding representation N_e demonstrates the authority score of the candidate and is generated by applying DeepWalk, node2vec, and ExEm methods on the collaborative network.

Note that BERTERS is a fixable method for finding experts in different environments and the difference emanates from the type of constructed graph. In order to apply BERTERS strategy into the CQAs, the desired graph is created based on the interactions between question posters and repliers. Other steps are done as described above.

4.4. Other features

Adding features results in having depth knowledge about candidates' expertise, accurately learning their subject areas, and improving precision. Since the h-index presents an author-level metric that evaluates both the fertility and citation impact of the publications of a researcher, BERTERS explores the h-index of candidates in terms of additional features and reputation score. The proper normalization of features is critical for convergence. So, the normalized value of h-index shown by Hi is combined with the features obtained from previous stages. In such a way, BERTERS creates expert representations using semantic representation and the structure information. Due to the capability of BERTERS in holding the authority, text similarity and reputation scores, it provides information-rich expert representations.

To use BERTERS in a CQA system, we can add the number of best answers provided by candidates, their reputation scores, number of thumbs up and down as extra features. It is clear that with this well-defined collection of information about candidates, BERTERS will be able to work properly.

4.5. Joint features

The important point in a multimodal deep learning model is to properly integrate multimodal features. But in practice, combining different modalities is challenging. Furthermore, modalities have different quantitative effects on predicting the outputs. There are at least three common ways to combine embedding vectors and create a single feature vector including summing, averaging, and concatenating [53]. In our case study, we select concatenation operator for two reasons. The first reason is that by concatenating features we will have a single representation that maintains three score values together without any changes. The other criterion is that the length of modality representations are not the same, hence it is not possible to use summing and averaging methods. Therefore, BERTERS integrates all features into a single representation through concatenation and gets $1 \times L$ vector, where L equals to the sum of the length of feature vectors.

In the next step, BERTERS employs a feed-forward neural network that consists of three stacked dense layers with ReLU activation function. The last layer is Sigmoid classifier. To efficiently train BERTERS, the cross-entropy loss is minimized and embeddings are learned jointly with all other model parameters. There is an important point in this step that should be noted. The proposed classifier can be replaced with any other classifier such as Random Forest, Support Vector Machine (SVM), and Logistic Regression.

5. Experiments

In this section, we will present the details of the experiment process. Firstly, we are going to present a summary of the datasets on which the BERTERS is applied. Then, we are going to describe the experimental setup of our work and the evaluation tasks, separately. Next, we will introduce the baseline algorithms to compare BERTERS against them. Later, we will provide an overview of model variations. In the end, the metrics hired to evaluate BERTERS will be specified.

Table 2
System Information.

	Model	Description
OS	Ubuntu 18.04.3 LTS	-
RAM	-	26G
CPU	Intel(R) Xeon(R)	2.20GHz
GPU	NVIDIA	Tesla P100

5.1. Dataset

In order to evaluate the quality of BERTERS in different environments, we used two various datasets. One dataset was gathered from Scopus to assess the performance of BERTERS in the academic domain. The other dataset obtained from Quora, which is a popular CQA, was used to show the efficiency and effectiveness of BERTERS in diverse scope and compare it with other related works. We will explain these datasets in the following paragraphs.

Scopus Dataset: To evaluate the performance of BERTERS, we investigated for a dataset that guarantees both content and non-content modalities. The dataset introduced in [20], gathered from Scopus, eliminates the requirement of a labeled data for constructing a collaborative network. The graph extracted from this dataset has arisen out of the collaborations of authors in different articles. Each node presents an author that his/her subject areas are considered as node labels. Moreover, the edges indicate the co-author interactions between authors. This dataset only ensures the data of graph modality. To adapt this dataset to our multimodal classification approach, we extracted other features from Scopus for text modality and extra feature. The obtained information consists of authors' articles and their h-index. While, the total number of the graph nodes is 27,473, we gathered the text information only for 9,378 authors.

Because BERTERS is a supervised multimodal classification approach, so it needs a ground truth for the learning part. To find a proper ground truth for our collected dataset, we followed the same procedure described in [20]. We derived a list of experts from Arnetminer for three topics: information extraction (IE), natural language processing (NLP), and machine learning (ML). This list of experts is defined as the ground truth. Figure 7 shows the word cloud presentation of the gathered articles related to the top expert in three topics.

Quora: This dataset was gathered by authors in study [43]. It includes the information of questions, answers, users and their following relationship in Twitter's social network. So, it's a suitable collection to test BERTERS in the CQA area. We treated the gathered data as done with study [44]. We considered all the answerers for each question and their received thumbs-up/down as expert candidates and the ground truth rating scores, respectively.

It should be noted that in both datasets our aim is not to predict the exact score value of each expert but to rank them according to their positions in the list. The descriptions of two datasets are summarized in Table 1.

5.2. Experimental setup

Table 2 presents the information of the system that the experiments were performed on. In our study, we employed a version of BERT called BERT-Small for learning the text representation. Its encoding and decoding parts have 4 stacked layers. Also, the size of the output embedding vector in BERT-Small is 512. On the other hand, SBERT provides different pre-trained models that perform well on one task, will show poor performance for other tasks. We used the model, called "stsb-roberta-large", that was optimized for Semantic Textual Similarity (STS). This model was trained on $SNLI + MultiNLI$ and then fine-tuned on the STS benchmark. The

(a) IE topic (b) ML topic (c) NLP topic

Fig. 7. Word cloud presentation of articles related to the top experts for three topics.

Table 3
Experimental setup.

USE embedding vector size	512
BERT embedding vector size	512
SBERT embedding vector size	1024
N_e vector size	128
h-index feature size	1
BERTERS$_{USE+N_e}$ size	641
BERTERS$_{BERT+N_e}$ size	641
BERTERS$_{SBERT+N_e}$ size	1153

output of this model is a 1024 dimensional vector. Moreover, there are several versions of USE models. We experimented with the last version (Version 5). Also, for node representation learning, we used the same parameters that are reported in [20]. The required information about setup is denoted in Table 3. In this table, BERTERS$_{SBERT+N_e}$, BERTERS$_{BERT+N_e}$ and BERTERS$_{USE+N_e}$ refer to BERTERS models use SBERT, BERT and USE independent of technique uses for node representation learning. Hence, we replace the name of the graph embedding technique with the output symbol N_e.

5.3. Tasks

We evaluated the performance of BERTERS on three tasks including multi-label classification, recommendation and visualization that are described in the following paragraphs.

5.3.1. Multi-label classification

In this task, the assumption is that each candidate is associated with one or more labels from a limited set L and the effort is to predict these labels with high precision. To conduct the multi-label classification task, we have a model that is trained with a portion of candidates and all their labels. It should be noted that the labels of candidates are defined according to their subject areas and represented as a one-hot numeric array. Then, a classification model gets the candidate vector representations to forecast the labels for the rest of the candidates. Different classifiers like Logistic Regression or SVM can be applied on a certain fraction of the expert embeddings whose subject areas are known. Then, the model predicts the subject areas for the remaining candidates. In other words, with the combination of expert embeddings and multi-label classification task, it is possible to anticipate the subject areas of experts for whom no specific information is available, and only their co-author connections with other experts are obtainable.

5.3.2. Recommendation

The task of recommendation is to suggest top K experts of interest to a given expert according to particular specifications such as similarity. Based on this task, the recommendation items are experts whose research interests and expertise are most similar to a given expert. This means that BERTERS ranks the experts for a given expert query according to their expert embeddings.

5.3.3. Visualization

Cluster visualization assists in the achievement of more vision into experts' cluster sets. Also, the low-dimensional vectors of experts present rich information about experts. Hence, BERTERS can illustrate the goodness of its embedding approach to cluster experts over three mentioned topics.

5.4. Baseline algorithms

In the succeeding paragraphs, we will compare BERTERS against the following baselines to approve the performance of BERTERS. Among them, TSPM, DRM, USE, BERT and SBERT convert the content features of candidates' expertise into low-dimensional vectors. While, ExpertsRank, AuthorityRank, ExEm$_{w2v}$, ExEm$_{ft}$, DeepWalk and Node2vec are graph-based that capture the non-content features of candidates from the collaborative network. On the other hand, GRMC, RMNL and MMSE are hybrid models that combine content and non-content features.

TSPM [54]: It is a topic-sensitive probabilistic model that finds experts by learning question representation via LDA-based model.

DRM [55]: It is also a topic-sensitive probabilistic model that obtains question representations via PLSA-based model to find experts in CQA.

USE [19]: This model converts the content features into low-dimensional vectors by transformer variant of USE.

BERT [17]: This model only operates on authors' articles. Each article is presented by a vector created form BERT transformer.

SBERT [18]: This model maps authors' articles into low-dimensional vectors using SBERT transformer.

ExpertsRank [56]: It is a PageRank-like algorithm used to calculate experts' scores in the user-user graph based on ask-answer relations of the users. ExpertsRank tries to find experts based on the degree of connections of experts with others in the collaborative network [20,25].

AuthorityRank [57]: It is an in-degree method that calculates user authority based on the number of best answers provided.

ExEm $_{ft}$ [20]: It is a version of ExEm that engages fastText method to learn the node representation.

ExEm $_{w2v}$ [20]: This one is another form of ExEm that creates vector representations for nodes by using Word2Vec.

DeepWalk [21]: This method represents a graph as a set of simple random walks starting on each node. Then, these random walks are trained using the skip-gram algorithm to create node embeddings [20].

Node2vec [22]: This approach is the modified version of DeepWalk with a more elaborate random walk. Node2vec proposed a biased-random walk using the breadth-first and depth-first search techniques [20].

GRMC [43]: It is a hybrid model that is created from both the social relationship between candidates and their history of questions and answers. In the proposed model, the goal is to consider expert finding as missing value estimation and estimate values via a matrix completion method.

RMNL [44]: Authors proposed a ranking metric network learning framework for the problem of the expert finding. They performed a heterogeneous CQA network built by the combination of both candidates' relative quality rank to questions and their social connections.

MMSE [45]: It is similar to GRMC. MMSE is a bayesian embedding model that integrates multiple modalities and multiple semantic perspectives. To deal with the multi-view property, authors utilized the Gaussian topic model to learn semantic embedding from both local view and global view. Also, they solved the sparse property of question-answer pairs by social structure information.

5.5. Model variations

We have experimented with several variants of the BERTERS model.

BERTERS$_{BERT+ExEm_{ft}}$: It is the combination of text and graph modalities. Text features are obtained by BERT transformer. On the other side, ExEm$_{ft}$ extracts node features from the co-author graph.

BERTERS$_{BERT+ExEm_{w2v}}$: Same as above but the node vectors are captured by ExEm$_{w2v}$.

BERTERS$_{BERT+Node2Vec}$: This architecture is almost identical to the previous one. The difference is that Node2Vec approach creates the node vectors.

BERTERS$_{BERT+DeepWalk}$: In this structure, DeepWalk derives the node features. The rest of the procedure is similar to the above.

BERTERS$_{USE+ExEm_{ft}}$: In this model, text features are obtained by USE model based on the transformer. Moreover, node features are extracted from the co-author graph using ExEm$_{ft}$.

BERTERS$_{USE+ExEm_{w2v}}$: This model learns text representation by the same structure outlined above, but it captures node vectors by ExEm$_{w2v}$.

BERTERS$_{USE+Node2Vec}$: In this structure, text features are presented by the transformer variation of USE. Also, the node vectors are derived by Node2Vec approach.

BERTERS$_{USE+DeepWalk}$: The procedure is similar to the above just DeepWalk obtains the nodes features.

BERTERS$_{SBERT+ExEm_{ft}}$: This model is the combination of SBERT and ExEm$_{ft}$ methods that learn text and node representations, respectively.

BERTERS$_{SBERT+ExEm_{w2v}}$: In this structure, SBERT and ExEm$_{w2v}$ methods are used to create the expert embeddings.

BERTERS$_{SBERT+Node2Vec}$: This model is the same as the previous and the only difference is using Node2Vec to capture the node features.

BERTERS$_{SBERT+DeepWalk}$: The model follows the same procedure described above just DeepWalk learns the node features.

5.6. Evaluation metrics

We use Micro-F1 and Macro-F1 scores as our metrics to assess the quality of BERTERS on the classification task. Also, to evaluate the performance of BERTERS over recommendation task, Normalized Discounted Cumulative Gain (nDCG) is used. These metrics are defined as follows:

F1 score depends on both the precision and recall and is defined as a weighted average of these two metrics. Eq. (7) expresses F1 score.

$$F1 = 2 \times \frac{Pr \times Re}{Pr + Re} \tag{7}$$

here Pr and Re denote precision and recall, respectively.

Micro-F1 underlines the common labels in the dataset by taking into account the equal importance for each instance. This means that this score computes the F1 score of the accumulated contributions of all labels [20]. Micro-F1 is calculated as following:

$$Micro - F1 = 2 \times \frac{microPr \times microRe}{microPr + microRe} \tag{8}$$

here *microPr* and *microRe* show the precision and recall in micro and Eq. (9) and (10) express their mathematical definitions.

$$microPr = \frac{\sum_{l \in L} TP_l}{\sum_{l \in L} (TP_l + FP_l)} \tag{9}$$

$$microRe = \frac{\sum_{l \in L} TP_l}{\sum_{l \in L} (TP_l + FN_l)} \tag{10}$$

where TP_l and FN_l present the number of true positives and false negatives within samples which are assigned to the label l [20].

Macro-F1 treats equally all labels to evaluate the overall performance of a classification model with regard to the common labels. The high value of Macro-F1 demonstrates that the model performs well on the rare labels.

$$Macro - F1 = \frac{\sum_{l \in L} F1(l)}{L} \tag{11}$$

here $F1(l)$ indicates the F1 score for label l.

nDCG evaluates the gold standard ranked list of experts against the ranked list outputs from recommendation task. The high value of nDCG show that there is a strong correlation between these two ranked lists. The DCG for k recommendations (DCG@k) sums the true scores ranked in the order induced by the predicted scores, meanwhile adding a logarithmic discount [20]. DCG@k is given by

$$DCG@k = ca_{rel_i} + \sum_{i=2}^{k} \frac{ca_{rel_i}}{log_2(i - 1 + 1)} = ca_{rel_i} + \sum_{i=2}^{k} \frac{ca_{rel_i}}{log_2(i)} \tag{12}$$

where ca_{rel_i} is the true relevance of the recommendation at position i for the current candidate ca. Then we can obtain nDCG@k as follow:

$$nDCG@k = \frac{DCG@k}{IDCG@k} \tag{13}$$

here IDCG is the DCG of ideal order.

6. Evaluation results

In the following paragraphs, firstly, we will investigate the efficiency of BERTERS variation models on the three tasks presented before. To makes more judgments about BERTERS on the classification and recommendation tasks, we are going to present results by varying the size of the training set. Then, we will examine the effect of the number of embedding dimensions on the performance.

6.1. Multi-label classification

One of the tasks for evaluating the performance of BERTERS is multi-label classification. A good expert embedding can be used as an input of a model that predicts the experts' labels. We used the Scopus dataset for the classification task because it is a labeled dataset that is suitable for the purpose of analyzing the capability of BERTERS in this task. We performed the classification task under three different scenarios. Firstly, we randomly selected a portion (10% to 90%) of candidates along with their labels as training data to evaluate BERTERS accomplishments on the remaining nodes. For this procedure, we trained a classifier with three fully connected layers. In the second scenario, we fixed the train ratio with a value of 50% and we made a broad comparison among four

Table 4
Micro-F1 of multi-label classification task varying the train-test split ratio.

Model	Train ratio								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
USE	0.5526	0.5696	0.5853	0.5943	0.5968	0.598	0.6004	0.6055	0.6063
BERT	0.5667	0.6058	0.618	0.612	0.6113	0.6088	0.6004	0.6294	0.6104
SBERT	0.5952	0.6343	0.6379	0.6379	0.6389	0.6381	0.6389	0.6389	0.6402
DeepWalk	0.5107	0.5297	0.5355	0.5413	0.5589	0.5552	0.5616	0.5627	0.5604
Node2vec	0.5123	0.5354	0.5406	0.5553	0.556	0.5601	0.5613	0.571	0.5693
$ExEm_{w2v}$	0.5256	0.5493	0.5458	0.561	0.5636	0.5637	0.5757	0.5851	0.5832
$ExEm_{ft}$	0.5222	0.5477	0.5455	0.5604	0.564	0.5656	0.5723	0.5864	0.5819
$BERTERS_{USE+DeepWalk}$	0.639	0.6459	0.6704	0.6797	0.6902	0.6992	0.6986	0.7115	0.7044
$BERTERS_{USE+Node2Vec}$	0.6351	0.6474	0.6652	0.6846	0.6868	0.6889	0.6992	0.6994	0.703
$BERTERS_{USE+ExEm_{w2v}}$	0.6338	0.6616	0.6765	0.6934	0.704	0.7011	0.7094	0.711	0.7124
$BERTERS_{USE+ExEm_{ft}}$	0.6358	0.6634	0.6847	0.6926	0.7011	0.7078	0.705	0.7155	0.7176
$BERTERS_{BERT+DeepWalk}$	0.6476	0.6613	0.6833	0.6884	0.6987	0.7032	0.7075	0.7045	0.7015
$BERTERS_{BERT+Node2Vec}$	0.6446	0.6618	0.6863	0.6881	0.6993	0.7042	0.7093	0.7023	0.7014
$BERTERS_{BERT+ExEm_{w2v}}$	0.6632	0.6785	0.6967	0.6921	0.7011	0.7117	0.7142	0.7081	0.7076
$BERTERS_{BERT+ExEm_{ft}}$	0.6595	0.6794	0.6927	0.6986	0.7099	0.7119	0.7127	0.7097	0.7091
$BERTERS_{SBERT+DeepWalk}$	0.6726	**0.6965**	0.7003	**0.7131**	0.7082	0.71	0.7111	0.7148	0.7123
$BERTERS_{SBERT+Node2Vec}$	0.6728	0.6898	0.7042	0.7045	**0.7149**	0.7089	0.7068	0.7154	0.7152
$BERTERS_{SBERT+ExEmw2v}$	0.6729	0.6933	0.7006	0.6963	0.7097	**0.7138**	**0.7162**	0.7128	0.718
$BERTERS_{SBERT+ExEm_{ft}}$	**0.6785**	0.6928	**0.7072**	0.7081	0.7006	0.7129	0.7111	**0.7163**	**0.7181**

Table 5
Macro-F1 of multi-label classification task varying the train-test split ratio.

Model	Train ratio								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
USE	0.4513	0.4679	0.5231	0.5169	0.5236	0.524	0.5261	0.5265	0.4945
BERT	0.4682	0.5187	0.5238	0.5299	0.5261	0.5284	0.5265	0.5292	0.5271
SBERT	0.4691	0.5255	0.532	0.5338	0.5301	0.536	0.5406	0.5319	0.5329
DeepWalk	0.3706	0.3941	0.3986	0.4109	0.4167	0.4218	0.4224	0.4302	0.4283
Node2vec	0.3651	0.3959	0.4004	0.4165	0.4154	0.4171	0.4218	0.431	0.4254
$ExEm_{w2v}$	0.3953	0.4005	0.4187	0.4219	0.422	0.4371	0.4378	0.4454	0.4387
$ExEm_{ft}$	0.3866	0.4019	0.4141	0.4215	0.4217	0.4338	0.4366	0.4521	0.4446
$BERTERS_{USE+DeepWalk}$	0.4978	0.5333	0.5474	0.5606	0.5771	0.5851	0.5841	0.5993	0.5781
$BERTERS_{USE+Node2Vec}$	0.4921	0.5332	0.538	0.5686	0.5686	0.5738	0.582	0.5773	0.5799
$BERTERS_{USE+ExEm_{w2v}}$	0.4999	0.5495	0.5514	0.5514	0.584	0.5939	0.5945	0.5938	0.5813
$BERTERS_{USE+ExEm_{ft}}$	0.5007	0.541	0.5687	0.5758	0.5884	0.5939	0.5936	0.5949	0.5957
$BERTERS_{BERT+DeepWalk}$	0.503	0.5481	0.5623	0.5708	0.5766	0.5842	0.5827	0.5817	0.5866
$BERTERS_{BERT+Node2Vec}$	0.5089	0.5555	0.565	0.5684	0.5799	0.5805	0.5812	0.5816	0.5793
$BERTERS_{BERT+ExEm_{w2v}}$	0.5185	0.5642	0.5782	0.5845	0.5853	0.5939	0.5963	0.5928	0.5909
$BERTERS_{BERT+ExEm_{ft}}$	0.5135	0.5647	0.5747	0.5828	0.5882	0.5919	0.5903	0.5943	0.5929
$BERTERS_{SBERT+DeepWalk}$	0.5159	0.5779	0.5892	0.5968	0.5973	0.5986	**0.6044**	0.5977	0.596
$BERTERS_{SBERT+Node2Vec}$	0.5386	0.5704	0.5913	0.592	**0.6052**	**0.6009**	0.5905	0.5999	**0.5978**
$BERTERS_{SBERT+ExEmw2v}$	0.5474	0.5752	0.5893	0.5817	0.5948	0.5942	0.5956	0.6004	0.5866
$BERTERS_{SBERT+ExEm_{ft}}$	**0.5544**	**0.5848**	**0.5952**	**0.5974**	0.5857	0.594	0.5961	**0.6015**	0.5899

classifiers including Random Forest, SVM, Logistic Regression, and a fully connected layer. In both first and second strategies, the dimensions of $BERTERS_{SBERT+N_e}$, $BERTERS_{BERT+N_e}$ and $BERTERS_{USE+N_e}$ equal 1153, 641 and 641, respectively. In the third scheme, we investigated the effect of a number of embedding dimensions on the performance of BERTERS by setting the train ratio with a value of 50% and changing the classifiers. Moreover, for the purpose of ensuring a fair comparison, we repeated the classification procedure 10 times and reported the results in terms of average Micro-F1 and average Macro-F1. In the paragraphs that follow, we are going to present the obtained results for each scenario.

6.1.1. First scenario: effect of train ratio

Tables 4 and 5 show the results of the classification task based on Micro-F1 and Macro-F1 scores for the first scenario under Scopus dataset. We separated the results into five groups. The first one illustrates single-modality based methods that use the content feature (published articles) to classify candidates. The second category denotes the approaches that employ graph embedding techniques to predict candidates' labels based on their co-author relationships. The rest of the groups are the multimodal approaches that organize based on their text representation learning part. According to the results of these tables we have the following observations:

i) It is conceivable that employing document embeddings built by SBERT presents better outcomes than embeddings obtained from other text representation learning methods.
ii) It is obvious that although ExEm, DeepWalk and Node2vec are random walk based methods, ExEm outperforms two other methods. The reason is that ExEm constructs intelligent random walks that comprise of at least two dominating nodes.
iii) Additionally, the results demonstrate that the learned embeddings from textual content using transformer models can better generalize over the classification task than the embeddings obtained from co-author network using graph embedding techniques.
iv) As well as, it can be concluded that two single-modality based methods are blamed for their poor performances in comparison with hybrid models.
v) It is evident that various versions of BERTERS gain the highest Micro-F1 and Macro-F1 scores. Given 90% of nodes as training data, as an example, $BERTERS_{SBERT+ExEm_{ft}}$ strengthens the performance on Micro-F1 metric by 12.16% and 23.40% compared

Table 6
Micro-F1 and Macro-F1 scores on the classification task for different classifiers (train ratio is 50%.)

Model	Classifier							
	Random Forest		SVM		Logistic Regression		Fully connected	
	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
$BERTERS_{USE+DeepWalk}$	0.7105	0.5282	0.696	0.5217	0.7026	0.557	0.29	0.1399
$BERTERS_{USE+Node2Vec}$	0.7107	0.5287	0.6941	0.5205	0.6998	0.5514	0.2931	0.1376
$BERTERS_{USE+ExEm_{w2v}}$	0.7106	0.5305	0.6972	0.525	0.7019	0.553	0.2971	0.146
$BERTERS_{USE+ExEm_{ft}}$	0.7112	0.5294	0.6984	0.5266	0.7028	0.559	0.2971	0.1447
$BERTERS_{BERT+DeepWalk}$	0.7168	0.5446	0.6706	0.5523	0.6848	0.5663	0.3133	0.1846
$BERTERS_{BERT+Node2Vec}$	0.7175	0.5474	0.6973	0.5693	0.698	**0.5728**	0.3609	0.1995
$BERTERS_{BERT+ExEm_{w2v}}$	**0.719**	0.5488	0.6706	0.5523	0.6848	0.5663	0.381	0.2225
$BERTERS_{BERT+ExEm_{ft}}$	0.717	0.545	0.6706	0.5523	0.6848	0.5663	0.4264	0.2098
$BERTERS_{SBERT+DeepWalk}$	0.7139	0.544	0.6729	0.5538	0.6836	0.5691	0.427	0.2703
$BERTERS_{SBERT+Node2Vec}$	0.7144	0.5458	0.6706	0.5523	0.6848	0.5663	0.3325	0.2389
$BERTERS_{SBERT+ExEmw2v}$	0.7139	0.5447	0.6733	0.5555	0.6838	0.5675	0.3875	0.2707
$BERTERS_{SBERT+ExEm_{ft}}$	0.7141	0.5456	0.6727	0.5562	0.686	0.5703	0.4355	0.2771

with SBERT and $ExEm_{ft}$ that are best in their groups. Moreover, $BERTERS_{SBERT+ExEm_{ft}}$ outperforms SBERT and $ExEm_{ft}$, as the best single-modality based methods, on Macro-F1 by 12.17% and 34.45% with the same amount of training data. These high values of gains show that our hypothesis about using multimodal and transfer learning, representing each candidate with a low-dimensional vector created from authority, text similarity and reparation scores, and obtaining better results is true.

vi) Also, we observe that among variants of BERTERS, $BERTERS_{SBERT+N_e}$ achieves high Micro and Macro values in all cases. It comes from the fact that SBERT is fine-tuned on NLI data, which generates semantically meaningful embeddings that considerably outperform BERT and USE. Moreover, the combination of SBERT and ExEm allows $BERTERS_{SBERT+ExEm}$ to exhibit a significant advantage over other BERTERS variation models.

vii) Further, the results suggest that train ratio is positive to the node classification performance. However, it has relatively little relevance to the performance of $BERTERS_{SBERT+N_e}$ and the differences are not that large in these cases. Shortly, according to the analysis, various models of $BERTERS_{SBERT+N_e}$ are not strictly sensitive to this parameter.

6.1.2. Second scenario: effect of classifier

Table 6 shows the results considering four classifiers trained on 50% Scopus dataset with $BERTERS_{USE+N_e}$, $BERTERS_{BERT+N_e}$ and $BERTERS_{SBERT+N_e}$ features. Note that we underline the best values of Micro and Macro scores for each classifier. Also, the bold numbers indicate the best results obtained overall. These experiments reveal a number of interesting points:

i) The results show that when the methods selected for the productions of the embeddings are BERT and $ExEm_{w2v}$, and the classifier is Random Forest, BERTERS has gained the highest Micro score among the competitors. Also, it is conspicuous that combination of Logistic Regression classifier and $BERTERS_{BERT+Node2Vec}$ approach yields the higher value of Macro score.

ii) In contrast to the first scenario that $BERTERS_{SBERT+N_e}$ produces the best outcomes using a fully connected classifier, $BERTERS_{BERT+N_e}$ outperforms other various versions of BERTERS over Random Forest and Logistic Regression classifiers in the second scenario. That means that the results of this experiment confirm that changing the classifier influences on the effectiveness of different models of BERTERS.

iii) It can be concluded that applying the Random Forest as a classifier on the expert embeddings extracted from different BERTERS's models makes a list of results in terms of Micro-F1 that are very closed to each other and show the highest scores compared to other classifiers.

iv) Using SVM as a classifier allows BERTERS generates the highest Macro score in comparison to Random Forest and fully connected classifiers.

6.1.3. Third scenario: effect of dimension

In the third scenario, we conducted investigations on the effect of embedding dimensions on the classification task. For this goal, we extracted the embeddings from the three fully connected layers, presented in Fig. 6, with sizes 256, 512 and 1024. Also, we fixed the train ratio with a value of 50%. Figure 8 illustrates the impacts of different embedding dimension sizes on various models of BERTERS over different classifiers. The observations from the results lead to the following conclusions:

i) The Micro score reaches a peak of 0.7472 by the combination of $BERTERS_{BERT+DeepWalk}$ and Random Forest in embedding size 512. Moreover, the combination of Logistic Regression and $BERTERS_{BERT+ExEm_{ft}}$ achieves the best performance in terms of Macro-F1.

ii) We find that Random Forest and the fully connected classifiers show the same trends by increasing the size of expert embeddings. Also, SVM and Logistic Regression follow a similar trend.

iii) When the embedding dimensions change from 256 to 512, we observe that the Micro and Macro scores have remained approximately constant in most cases.

iv) In contrast, we see the downtrend trends in all cases in the performance of BERTERS techniques by varying the number of dimensions from 512 to 1024.

v) Overall, we observe from this figure and Table 6 that using the expert embeddings created from $BERTERS_{SBERT+N_e}$, $BERTERS_{BERT+N_e}$ and $BERTERS_{USE+N_e}$ with 1153, 641 and 641 dimensions, respectively, allows BERTERS to exhibit significant advantages over other embedding sizes.

6.2. Recommendation

The purpose of this experiment is to show how BERTERS models can be effectively used to order expert recommendations with the help of the learned expert embeddings. To judge BERTERS performance on various benchmarks and areas, we evaluated the low-dimensional vectors created from BERTERS over Quora and Scopus datasets. The goal of expert finding in Quora dataset is to return a ranked list of experts with expertise on a given question. By way of explanation, the target is to route a newly posted question to users that are experts on the topic of the question. While in Scopus dataset, we have the desire to recommend potential research collaborator in a distinct topic.

Table 7
Summary of previous studies.

Study	Model			Methodology	Cite	Year
	Document-based	Graph-based	Hybrid			
ExpertsRank [56]	-	✓	-	Link analysis	1111	2007
AuthorityRank [57]	-	✓	-	Link analysis	283	2008
CQARank [41]	-	-	✓	Link analysis, topic model	225	2013
Riahi et al. [27]	✓	-	-	TF-IDF, LDA, STM	209	2012
ExpertRank [8]	-	-	✓	Link analysis, TF-IDF	208	2013
TSPM [54]	✓	-	-	LDA	199	2008
GRMC [43]	-	-	✓	Matrix completion	142	2014
Fu et al. [35]	-	✓	-	Propagation	101	2007
Liu et al. [10]	-	-	✓	Link analysis, TF-IDF, reputation score	95	2013
RMNL [44]	-	-	✓	LSTM, DeepWalk	75	2016
HSNL [42]	-	-	✓	LSTM, graph embedding	66	2016
Neshati et al. [29]	✓	-	-	LDA, language model	53	2017
Zhou et al. [13]	-	-	✓	Link analysis, LDA, reputation score	49	2014
DRM [55]	✓	-	-	PLSA	42	2013
Momtazi and Naumann [28]	✓	-	-	LDA	37	2013
Wang et al. [33]	✓	-	-	Word2Vec, CNN	26	2017
Sun et al. [36]	-	✓	-	Graph embedding	22	2019
Li et al.[30]	✓	-	-	LDA	17	2015
Expert2Vec [11]	-	-	✓	Word2Vec,reputation score	13	2019
ExpFinder [46]	-	-	✓	TF-ID, Link analysis	10	2021
Dehghan et al.[34]	✓	-	-	LSTM	9	2019
Xie et al. [40]	-	-	✓	Link analysis, LDA	6	2016
MMSE [45]	-	-	✓	Word2Vec, DeepWalk	6	2019
Mumtaz and Wang [38]	-	✓	-	Betweenness centrality	5	2017
Doc2vec [32]	✓	-	-	Doc2vec	3	2020
de Campos et al. [31]	✓	-	-	LDA	1	2021
ExEm [20]	-	✓	-	Graph embedding	1	2021

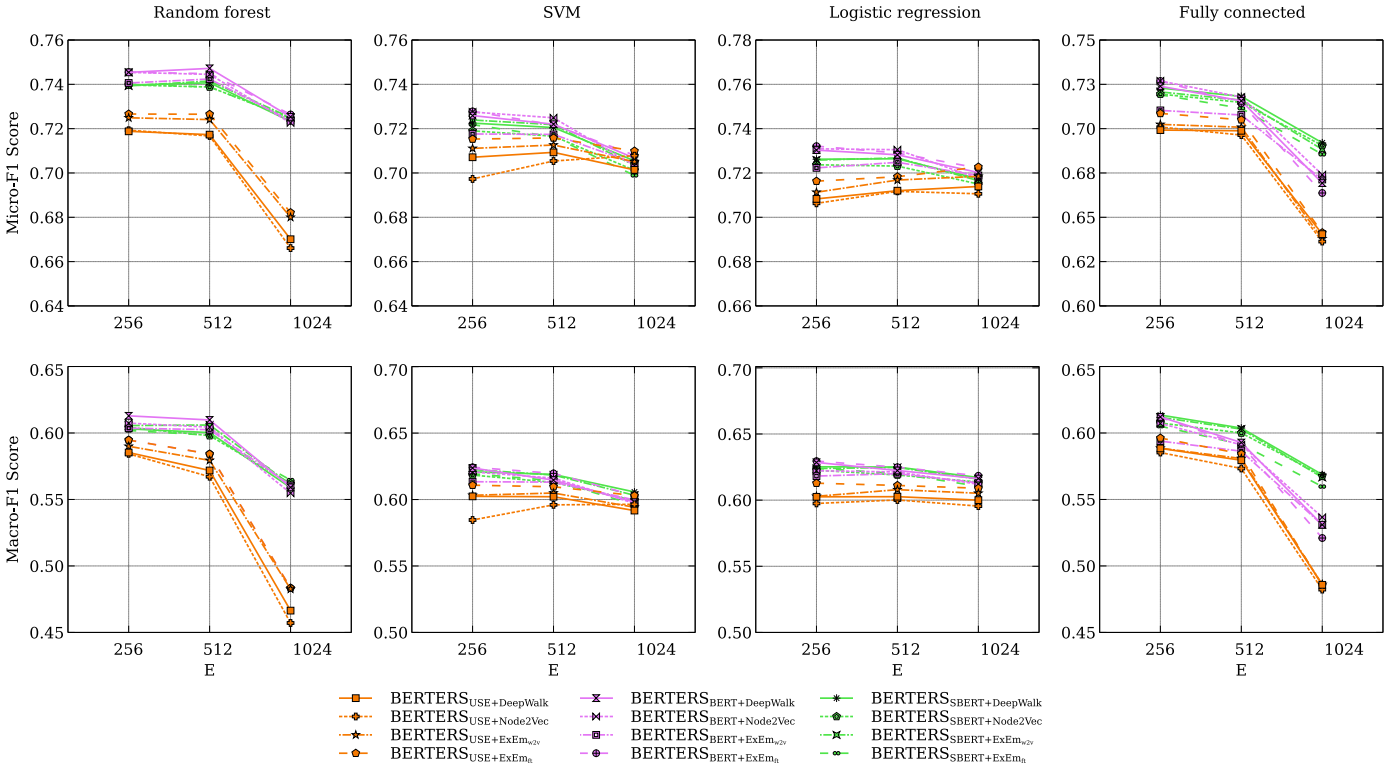


Fig. 8. Micro-F1 and Macro-F1 scores on the classification task for different classifiers over the diverse number of dimensions (train ratio is 50%).

In Quora dataset, we embedded questions using SBERT transformer with dimension 1024. Besides, since the vectors should always be the same length for doing cosine similarity, we extracted the embeddings of candidates from the first layer of neural network in Fig. 6.

In Scopus dataset, we selected three topics: IE, NLP and ML. The lists of people in these topics are used as experts to construct the ground truth to evaluate the recommendation task. We picked the top experts in each topic as the query node and used cosine similarity to measure the distance between the expert embedding vectors and the query node. We recommended the nearest candidates to the query as experts.

It should be noted that our task in both datasets is not to predict the exact score value of each expert but to rank them in terms

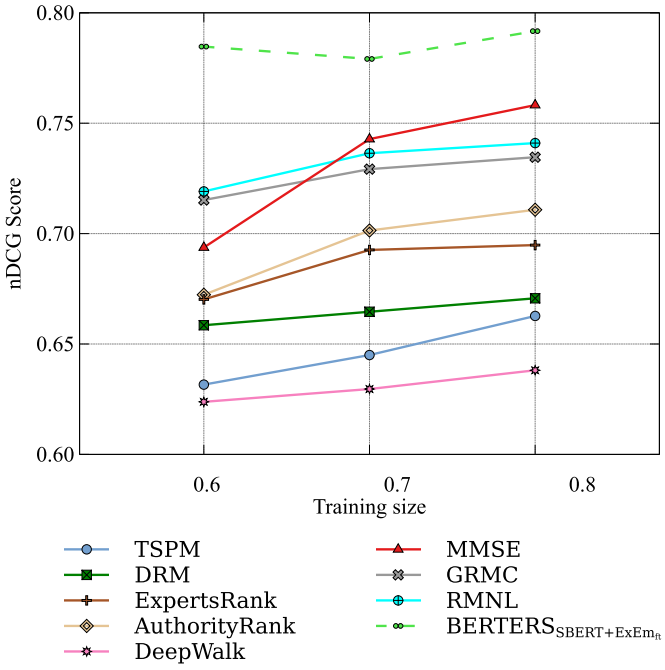


Fig. 9. Experimental results on nDCG with different proportions of Quora dataset for training.

of their positions in the list. On the other hand, we accommodated the position of the experts in these lists as their ranks for the ground truth.

We have two scenarios for the recommendation task. Moreover, the first outline consists of two sections. In one section we compared BERTERS$_{SBERT+ExEm_{ft}}$ with the state-of-the-art models including TSPM, DRM, ExpertsRank, AuthorityRank, GRMC, RMNL and MMSE on Quora benchmark. In the second part, we reported the results on Scopus to show the effect of the number of recommended experts in terms of nDCG@k where k was set to 6, 7 and 8. Finally, in the second scheme, we investigated the effect of embedding dimensions on Scopus dataset.

6.2.1. First scenario: effect of number of recommended experts at k

Quora: The performance of different models in terms of nDCG is summarised and reported in Fig. 9. The DeepWalk, AuthorityRank and ExpertsRank methods are graph-based ones while TSPM and DRM methods are document-based models. Also, GRMC, RMNL and MMSE are hybrid approaches that consider both link analysis of users and their published contents. This experiment reveals a number of interesting points:

i) Contrary to our expectations, two graph-based models AuthorityRank and ExpertsRank outperform the document-based methods, TSPM and DRM.
ii) DeepWalk that learns the vector representations of candidates from the graph gives the poorest results.
iii) The hybrid methods, GRMC, RMNL, MMSE and BERTERS$_{SBERT+ExEm_{ft}}$ outperform single-modality based methods.
iv) In all the cases, our BERTERS$_{SBERT+ExEm_{ft}}$ model achieves the best performance. BERTERS$_{SBERT+ExEm_{ft}}$ works better than the baselines with gains of 9.12%, 4.88% and 4.41% with regards to the training sizes. This fact shows that leveraging the power of creating single representations for candidates from their authority, text similarity and reparation scores, instead of calculating separate scores and merging them, can further improve the performance of expert finding in the question answering system.

Scopus: Fig. 10 demonstrates nDCG score provided by the identified top k experts in three specific topics over Scopus dataset. As can be seen, we compared different models of BERTERS with each other. Results in this figure lead to the following observations:

i) BERTERS$_{USE+ExEm_{ft}}$, BERTERS$_{USE+Node2Vec}$ and BERTERS$_{USE+DeepWalk}$ outperform other BERTERS techniques in NLP, IE and ML topics, accordingly.
ii) As the number of recommendation increases, the performance of all methods go up linearly.
iii) In contrast to the classification task, BERTERS$_{SBERT+N_e}$ performs poorly in the recommendation process. Also, different versions of this model present very closed outputs.

6.2.2. Second scenario: effect of dimension

In this strategy, we studied the effect of embedding dimensions in terms of nDCG@8 score on only three methods BERTERS$_{USE+ExEm_{ft}}$, BERTERS$_{BERT+ExEm_{ft}}$ and BERTERS$_{SBERT+ExEm_{ft}}$ for Scopus dataset. We used the output embeddings created from three dense layers of the proposed model in Fig. 6 with 256, 512 and 1024 dimensions. Figure 11 illustrates the effect of embedding dimensions on BERTERS models. The following considerations are made from this scenario:

i) In NLP topic, BERTERS$_{USE+ExEm_{ft}}$ achieves the best score with the embedding size 1024. In IE topic, BERTERS$_{SBERT+ExEm_{ft}}$ has gained the highest value among the competitors. Both BERTERS$_{USE+ExEm_{ft}}$ and BERTERS$_{BERT+ExEm_{ft}}$ are the winners in ML topic.
ii) In opposition to the classification task, the performance of BERTERS models decreases from 256 to 512, and the scores rise from 512 to 1024 dimensions.
iii) A comparison between the results of Figs. 11 and 10 shows that changing the size of the expert embeddings makes additional gains for BERTERS approaches in NLP and ML topics. While resizing the expert vectors does not have any specific effect on the performance of BERTERS in IE subject.

6.3. Visualization

The goal of this task is to show that BERTERS is able to cluster together experts in the same field. For visualisation purpose, there are a number of dimension reduction techniques like Principal Component Analysis (PCA), and Uniform Manifold Approximation and Projection (UMAP) that can be applied on low-dimensional vectors. To interpret expert embeddings using visualization technique, we used Scopus dataset, chose 50 top experts in each topic and visualized them based on their embeddings using UMAP. As Fig. 12 shows three different topics are used to highlight the experts in different colors. Figure 12(a-c) cluster experts based on BERTERS$_{USE+ExEm_{w2v}}$, BERTERS$_{BERT+ExEm_{w2v}}$ and BERTERS$_{SBERT+ExEm_{w2v}}$, respectively. Also, we can see that we compared BERTERS with SBERT and ExEm$_{w2v}$ (Fig. 12d and e) which are best in their groups and use text and graph modalities, accordingly. From the figures, we have the following observations:

i) Among different versions of BERTERS tested in this experiment, BERTERS$_{BERT+ExEm_{w2v}}$ and BERTERS$_{SBERT+ExEm_{w2v}}$ well separate the communities. Also, the experts of each cluster are well distinguished by these two methods and this illustrates the power of their embeddings. Moreover, the partitiones originated by these approaches are more meaningful. The reason is that three topics have overlaps, and a candidate can be expert in all of them. So, the vector embeddings created by these approaches allow to compare expert candidates and find clusters of experts that are similar in characteristics.

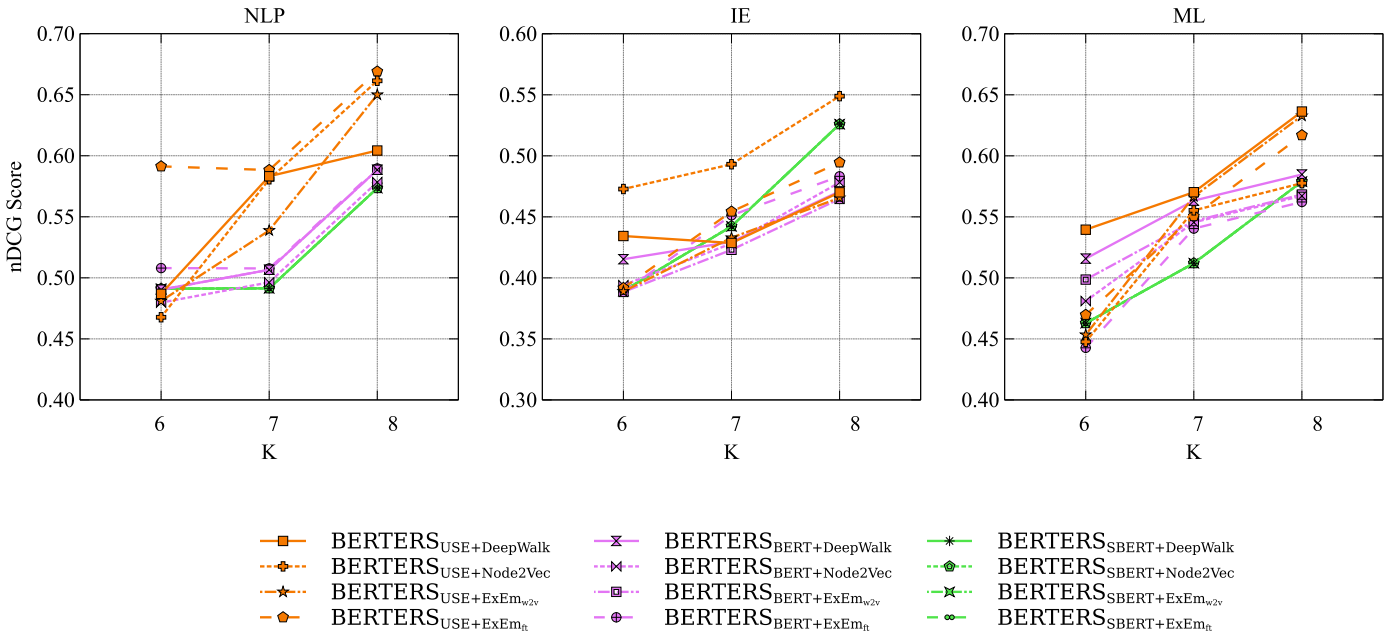


Fig. 10. Experimental results on nDCG based on top k experts of Scopus dataset.

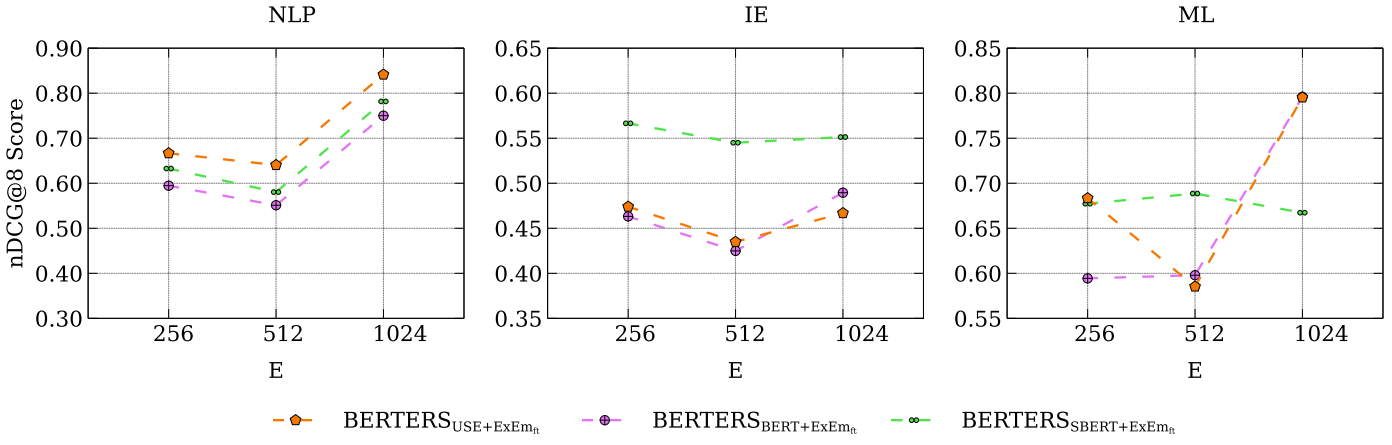


Fig. 11. nDCG@8 score for three topics with varying dimensions.

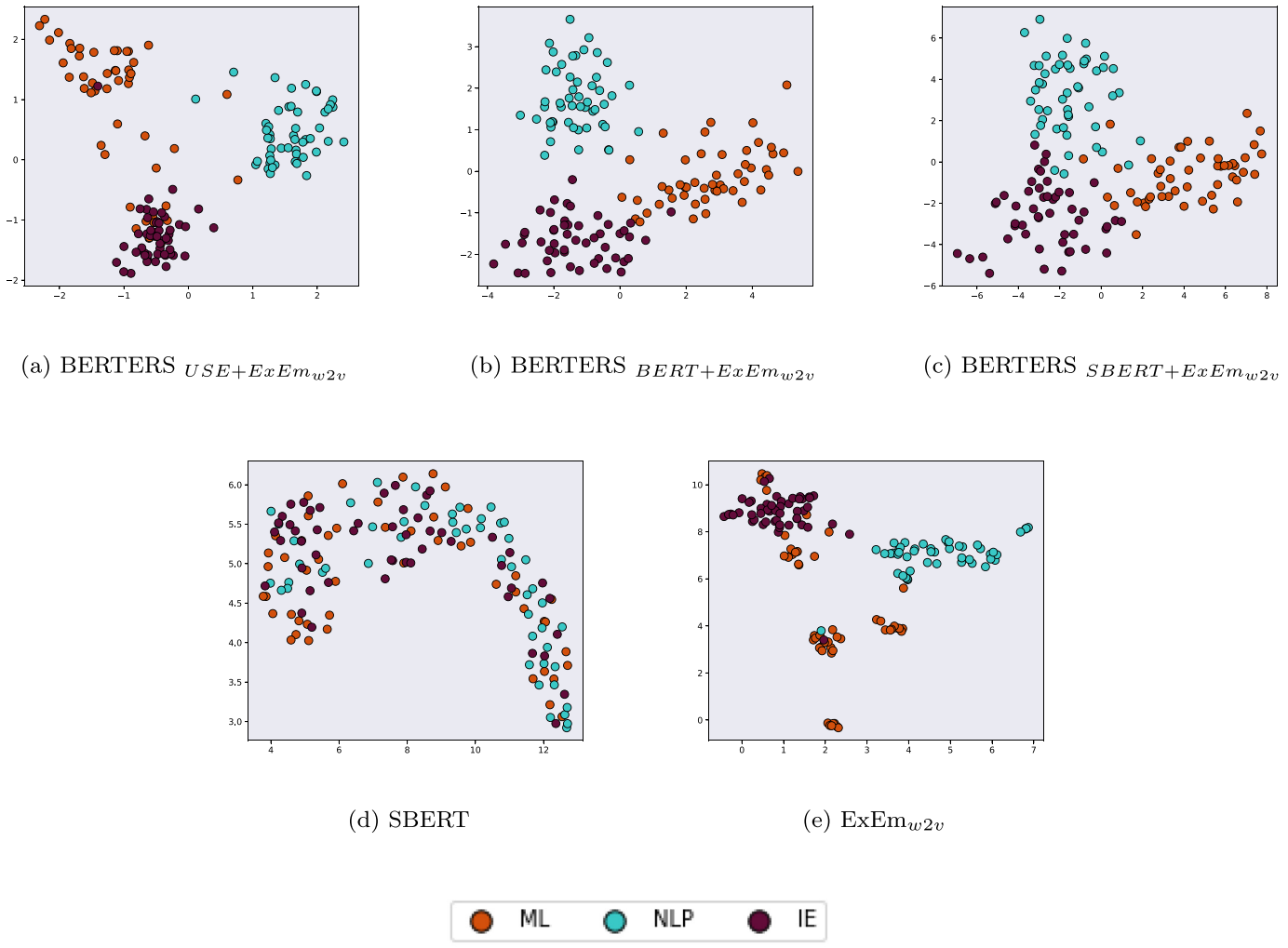
ii) While BERTERS$_{USE+ExEm_{w2v}}$ shows poor performance compared to BERTERS$_{BERT+ExEm_{w2v}}$ and BERTERS$_{SBERT+ExEm_{w2v}}$, it significantly outperforms SBERT and ExEm$_{w2v}$.
iii) In opposition to BERTERS approaches, SBERT embeds communities very closely. Since the contents of articles related to these experts are so similar, SBERT cannot make a distinction between the expertise of these experts.
iv) Using ExEm$_{w2v}$ without content data yields to completely separating three subject areas of ML, NLP and IE and we know that it is not correct due to the fact that there are significant overlaps among topics. Also, ExEm$_{w2v}$ fails in separating the experts in a cluster.

7. Discussion

The main idea behind using expert embeddings to find similar experts is that of the distributional hypothesis. Fundamentally, the distributional hypothesis declares that experts that appear in similar text and graph contexts are more likely to be similar to each other. Previous proposed hybrid models individually computed authority, text similarity and reputation scores and then merged them by a combination strategy. Based on the application scenario, these studies considered higher priority for each score. While the main objective of this research was to address the problem of combinations of different scores by proposing a multimodal and transfer learning-based approach. The suggested model presents each expert candidate by a single vector representation that originates from the text content, user relationship information and other features.

In this research, we compared the candidate vectors produced by BERTERS, single-modality based methods and hybrid approaches on a variety of tasks and various benchmarks to see which produced the most accurate candidate vectors. Furthermore, we investigated the effects of dimensions on BERTERS approaches. This evaluation would help provide insight into that BERTERS methods perform better in which dimensions.

One test used to determine this accuracy was the multi-label classification task. The results of Micro and Macro scores illustrated that different versions of BERTERS produced the most accurate expert vectors than single-modality based methods. Moreover, we see that among the BERTERS approaches, BERTERS$_{SBERT+ExEm}$ is superior to others in terms of metrics in this task. This is due to the combination of SBERT and ExEm allows BERTERS$_{SBERT+ExEm}$ to exhibit a significant advantage over other BERTERS variation models.



(a) BERTERS $_{USE+ExEm_{w2v}}$ (b) BERTERS $_{BERT+ExEm_{w2v}}$ (c) BERTERS $_{SBERT+ExEm_{w2v}}$

(d) SBERT (e) ExEm$_{w2v}$

Fig. 12. Visualization of communities of 50 top experts in three topics for different techniques and dimensions. Each point corresponds to an expert. Color of an expert denotes its cluster.

Additionally, we compared BERTERS with single-modality based methods and hybrid approaches in a recommendation analysis over two datasets in order to determine which was the most effective at recommending the accurate experts in a certain topic. BERTERS$_{SBERT+ExEm_{ft}}$ improved the performance of expert finding by taking advantage of ExEm as a graph embedding technique to create node representation and combine it with deep representation of contents learned from SBERT transformer. In this way, the candidate embedding produced by BERTERS$_{SBERT+ExEm_{ft}}$ contains rich semantic and syntactic information of candidate. Therefore, in this examination, the candidate embeddings by BERTERS$_{SBERT+ExEm_{ft}}$ was the highly accurate that produced meaningful results.

Also, final test to measure the quality of the BERTERS embeddings was the visualization. The results verify that BERTERS$_{BERT+ExEm_{w2v}}$ and BERTERS$_{SBERT+ExEm_{w2v}}$ group experts in a way that similarity between experts that belong to the same topic becomes as high as possible, while similarity between experts from different topics gets as small as possible. Having this in mind that there are authors who have been working in multiple fields such as IE and NLP together or any other combination of these three subject areas. A clear separation in this presentation is not always acceptable and for some hard cases such as what is shown in the results, the inseparable subject areas must have collisions in some cases.

As a conclusion, there are several reasons for the success of BERTERS. One of them originates from the usage of multimodal and transfer learning approaches. The second reason behind this superiority is because of selecting the features that better characterize candidates. The other motivation is presenting candidates as low-dimensional vectors created by recent researches in both text and graph representations such as transformers and graph embeddings. Finally, the best embeddings for finding experts are directly generated from the concatenation of their values of normalized h-index, their presentations obtained from a co-author network by ExEm and candidates' published items that converted into vectors by transformers. Also, the results prove the capability of BERTERS to extend into a variety of domains and areas such as CQA to find the best users for answering the posted questions.

In the future, we expect to further investigate the temporal aspect of candidates. Since the interests and expertise of candidates vary over time, so obtaining the temporal information of candidates makes it more applicable to the recommendation task in the real world. Another interesting aspect is to extend BERTERS into other domains. Although these results provide some insight into the effectiveness of BERTERS performance in academic and CQA domains, there are still other areas that this work could be tested upon in the future. For instance, finding reviewers who are recognized as having a higher level of expertise in online business review systems. Customers perceive a review as more useful when

reviewers provide high quality reviews. In this context, the features of candidates and the ways of representing them might be changed. Therefore, it is an interesting problem to analyze these components and the ways of obtaining information from them.

8. Conclusion

In this paper, a multimodal and transfer learning-based approach, called BERTERS, has been proposed for an expert recommendation system. BERTERS presented each expert candidate by a single vector representation that shows the authority, text similarity, and reputation scores. BERTERS directly used both transformers and the graph embedding techniques to convert the content and non-content information into low-dimensional vectors. Furthermore, BERTERS added other extra features like reputation score to obtain the final representation. The proposed expert embeddings can benefit a lot of applications such as expert classification, expert clustering, expert recommendation, detecting communities of experts, link prediction. We evaluated the performance of BERTERS in classification, recommendation and visualisation tasks. The results demonstrated that the hypothesis about using multimodal and transfer learning, representing each candidate with a low-dimensional vector created from authority, text similarity and reparation scores, and obtaining better results is true.

Declaration of Competing Interest

None.

CRediT authorship contribution statement

N. Nikzad-Khasmakhi: Conceptualization, Methodology, Validation, Investigation, Data curation, Software, Writing – original draft. **M.A. Balafar:** Project administration, Conceptualization, Formal analysis, Methodology, Validation. **M. Reza Feizi-Derakhshi:** Supervision, Investigation. **Cina Motamed:** Writing – review & editing, Validation.

References

[1] Altan A, Karasu S. Recognition of COVID-19 disease from X-ray images by hybrid model consisting of 2D curvelet transform, chaotic salp swarm algorithm and deep learning technique. Chaos Solitons Fractals 2020;140:110071.

[2] Altan A, Karasu S, Zio E. A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. Appl Soft Comput 2021;100:106996.

[3] Zhang S, Yao L, Sun A, Tay Y. Deep learning based recommender system: a survey and new perspectives. 2019. arXiv:1707.07435. 10.1145/3285029

[4] Bobadilla J, Ortega F, Hernando A, Gutiérrez A. Recommender systems survey. Knowl-Based Syst 2013. doi:10.1016/j.knosys.2013.03.012.

[5] Isinkaye F.O., Folajimi Y.O., Ojokoh B.A.. Recommendation systems: principles, methods and evaluation. 2015. 10.1016/j.eij.2015.06.005

[6] Zhen L, Song HT, He JT. Recommender systems for personal knowledge management in collaborative environments. Expert Syst Appl 2012. doi:10.1016/j.eswa.2012.04.060.

[7] Nikzad-Khasmakhi N, Balafar M, Feizi-Derakhshi MR. The state-of-the-art in expert recommendation systems. Eng Appl Artif Intell 2019;82:126–47.

[8] Wang GA, Jiao J, Abrahams AS, Fan W, Zhang Z. ExpertRank: a topic-aware expert finding algorithm for online knowledge communities. Decis Support Syst 2013. doi:10.1016/j.dss.2012.12.020.

[9] Yuan S, Zhang Y, Tang J, Cabotà JB. Expert finding in community question answering: a review. CoRR 2018. abs/1804.07958.

[10] Liu DR, Chen YH, Kao WC, Wang HW. Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. Inf Process Manage 2013. doi:10.1016/j.ipm.2012.07.002.

[11] Mumtaz S, Rodriguez C, Benatallah B. Expert2Vec: experts representation in community question answering for question routing. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics); 2019. ISBN 9783030212896. doi:101007/978-3-030-21290-2_14.

[12] Yang KH, Chen CY, Lee HM, Ho JM. EFS: Expert finding system based on wikipedia link pattern analysis. In: Conference proceedings - IEEE international conference on systems, man and cybernetics; 2008.

[13] Zhou G, Zhao J, He T, Wu W. An empirical study of topic-sensitive probabilistic model for expert finding in question answer communities. Knowl-Based Syst 2014. doi:10.1016/j.knosys.2014.04.032.

[14] Ganesh J, Ganguly S, Gupta M, Varma V, Pudi V. Author2vec: learning author representations by combining content and link information. WWW (Companion volume); 2016.

[15] Karasu S, Altan A, Bekiros S, Ahmad W. A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series. Energy 2020;212:118750.

[16] Dosovitskiy A, Djolonga J. You only train once: loss-conditional training of deep networks. In: International conference on learning representations; 2019.

[17] Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understandingp ;2018 arXiv:181004805.

[18] Reimers N, Gurevych I. Sentence-bert: sentence embeddings using siamese bert-networks ;2019 arXiv:190810084.

[19] Cer D, Yang Y, Kong S-y, Hua N, Limtiaco N, John RS, Constant N, Guajardo-Cespedes M, Yuan S, Tar C, et al. Universal sentence encoder ;2018 arXiv:180311175.

[20] Nikzad-Khasmakhi N, Balafar M, Feizi-Derakhshi MR, Motamed C. Exem: expert embedding using dominating set theory with deep learning approaches. Expert Syst Appl 2021;177:114913.

[21] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining; 2014 1403.6652. ISBN 9781450329569

[22] Grover A, Leskovec J. Node2vec: scalable feature learning for networks. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining; 2016 arXiv:1607.00653. ISBN 9781450342322

[23] Karasu S, Altan A. Recognition model for solar radiation time series based on random forest with feature selection approach. In: 2019 11th International conference on electrical and electronics engineering (ELECO). IEEE; 2019. p. 8–11.

[24] Altan A, Karasu S. The effect of kernel values in support vector machine to forecasting performance of financial time series. J Cognit Syst 2019;4(1):17–21.

[25] Lin S, Hong W, Wang D, Li T. A survey on expert finding techniques. J Intell Inf Syst 2017. doi:10.1007/s10844-016-0440-5.

[26] Yuan S, Zhang Y, Tang J, Hall W, Cabotà JB. Expert finding in community question answering: a review. Artif Intell Rev 2020;53(2):843–74.

[27] Riahi F, Zolaktaf Z, Shafiei M, Milios E. Finding expert users in community question answering. In: Proceedings of the 21st international conference on world wide web. In: WWW '12 Companion. New York, NY, USA: Association for Computing Machinery; 2012. p. 791–8. doi:10.1145/2187980.2188202. ISBN 9781450312301

[28] Momtazi S, Naumann F. Topic modeling for expert finding using latent Dirichlet allocation. WIREs Data Min. Knowl. Discov. 2013. doi:10.1002/widm.1102.

[29] Neshati M, Fallahnejad Z, Beigy H. On dynamicity of expert finding in community question answering. Inf Process Manage 2017. doi:10.1016/j.ipm.2017.04.002.

[30] Li H, Jin S, Li S. A hybrid model for experts finding in community question answering. In: Proceedings - 2015 international conference on cyber-enabled distributed computing and knowledge discovery, CyberC 2015; 2015. ISBN 9781467391993

[31] de Campos LM, Fernández-Luna JM, Huete JF, Redondo-Expósito L. Lda-based term profiles for expert finding in a political setting. J Intell Inf Syst 2021:1–31.

[32] Rampisela TV, Yulianti E. Academic expert finding in indonesia using word embedding and document embedding: a case study of fasilkom UI. In: 2020 8th International conference on information and communication technology (ICoICT). IEEE; 2020. p. 1–6.

[33] Wang J, Sun J, Lin H, Dong H, Zhang S. Convolutional neural networks for expert recommendation in community question answering. Sci China Inf Sci 2017;60(11):1–9.

[34] Dehghan M, Biabani M, Abin AA. Temporal expert profiling: with an application to t-shaped expert finding. Inf Process Manage 2019;56(3):1067–79.

[35] Fu Y, Xiang R, Liu Y, Zhang M, Ma S. Finding experts using social network analysis. In: Proceedings of the IEEE/WIC/ACM international conference on web intelligence, WI 2007; 2007. ISBN 0769530265

[36] Sun J, Bandyopadhyay B, Bashizade A, Liang J, Sadayappan P, Parthasarathy S. ATP: directed graph embedding with asymmetric transitivity preservation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33; 2019. p. 265–72.

[37] Zhan J, Guidibande V, Parsa SPK. Identification of top-K influential communities in big networks. J Big Data 2016. doi:10.1186/s40537-016-0050-7.

[38] Mumtaz S, Wang X. Identifying top-k influential nodes in networks. In: Proceedings of the 2017 ACM on conference on information and knowledge management; 2017. p. 2219–22.

[39] Bian R., Koh Y.S., Dobbie G., Divoli A.. Identifying top-k nodes in social networks: a survey. 2019. 10.1145/3301286

[40] Xie X, Li Y, Zhang Z, Pan H, Han S. A topic-specific contextual expert finding method in social network. In: Asia-pacific web conference. Springer; 2016. p. 292–303.

[41] Yang L, Qiu M, Gottipati S, Zhu F, Jiang J, Sun H, Chen Z. CQARank: jointly model topics and expertise in community question answering. In: International conference on information and knowledge management, proceedings; 2013. ISBN 9781450322638

[42] Fang H, Wu F, Zhao Z, Duan X, Zhuang Y, Ester M. Community-based question answering via heterogeneous social network learning. In: Proceedings of the AAAI conference on artificial intelligence, vol. 30; 2016.

[43] Zhao Z, Zhang L, He X, Ng W. Expert finding for question answering via graph regularized matrix completion. IEEE Trans Knowl Data Eng 2015. doi:10.1109/TKDE.2014.2356461.
[44] Zhou Z, Qifan Y, Cai D, He X, Yueting Z. Expert finding for community-based question answering via ranking metric network learning. In: IJCAI International joint conference on artificial intelligence; 2016.
[45] Sang L, Xu M, Qian SS, Wu X. Multi-modal multi-view Bayesian semantic embedding for community question answering. Neurocomputing 2019. doi:10.1016/j.neucom.2018.12.067.
[46] Kang Y, Du H, Forkan ARM, Jayaraman PP, Aryani A, Sellis T. Expfinder: an ensemble expert finding model integrating n-gram vector space model and μcohits. CoRR 2021. abs/2101.06821.
[47] Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J. Deep learning–based text classification: a comprehensive review. ACM Comput Surv (CSUR) 2021;54(3):1–40.
[48] Sun F, Liu J, Wu J, Pei C, Lin X, Ou W, Jiang P. BERT4Rec: sequential recommendation with bidirectional encoder representations from transformer. In: International conference on information and knowledge management, proceedings; 2019 arXiv:1904.06690. ISBN 9781450369763
[49] Asgari-Chenaghlu M., Feizi-Derakhshi M.R., Farzinvash L., Motamed C.. A multimodal deep learning approach for named entity recognition from social media. arXiv preprint arXiv:200106888 2020;.
[50] Nettleton D.F.. Data mining of social networks represented as graphs. 2013. 10.1016/j.cosrev.2012.12.001
[51] Cai H, Zheng VW, Chang KCC. A comprehensive survey of graph embedding: problems, techniques, and applications. IEEE Trans Knowl Data Eng 2018. arXiv:1709.07604. doi:10.1109/TKDE.2018.2807452.
[52] Goyal P, Ferrara E. Graph embedding techniques, applications, and performance: a survey. Knowl-Based Syst 2018. arXiv:1705.02801. doi:10.1016/j.knosys.2018.03.022.
[53] Damoulas T, Girolami MA. Combining feature spaces for classification. Pattern Recognit 2009. doi:10.1016/j.patcog.2009.04.002.
[54] Guo J, Xu S, Bao S, Yu Y. Tapping on the potential of Q&A community by recommending answer providers. In: Proceedings of the 17th ACM conference on information and knowledge management; 2008. p. 921–30.
[55] Hashemi SH, Neshati M, Beigy H. Expertise retrieval in bibliographic network: a topic dominance learning approach. In: Proceedings of the 22nd ACM international conference on information & knowledge management; 2013. p. 1117–26.
[56] Zhang J, Ackerman MS, Adamic L. Expertise networks in online communities: structure and algorithms. In: Proceedings of the 16th international conference on world wide web; 2007. p. 221–30.
[57] Bouguessa M, Dumoulin B, Wang S. Identifying authoritative actors in question-answering forums: the case of yahoo! answers. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining; 2008. p. 866–74.
[58] Elahe Nasiri, et al. A new link prediction in multiplex networks using topologically biased random walks. Chaos, Solitons & Fractals 2021 In press. doi:10.1016/j.chaos.2021.111230.
[59] Kamal Berahmand, et al. A preference random walk algorithm for link prediction through mutual influence nodes in complex networks. Journal of King Saud University Computer and Information Sciences 2021.
[60] Saman Forouzandeh, et al. Presentation of a recommender system with ensemble learning and graph embedding: a case on MovieLens. Multimedia Tools and Applications 2021.
[61] Meysam Asgari-Chenaghlu. TopicBERT: A cognitive approach for topic detection from multimodal post stream using BERT and memory–graph. Chaos, Solitons & Fractals 2021.