IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 3, 2020, accepted July 12, 2020, date of publication July 15, 2020, date of current version July 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009393

Combination of ELMo Representation and CNN Approaches to Enhance Service Discovery

ZHAO HUANG[1,2] AND WEI ZHAO[2]
[1]Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710062, China
[2]School of Computer Science, Shaanxi Normal University, Xi'an 710119, China

Corresponding author: Wei Zhao (weizhao@snnu.edu.cn)

This work was supported in part by the Research Grants funded by the National Natural Science Foundation of China under Grant 61771297, and in part by the Fundamental Research Funds for the Central Universities under Grant GK201802013.

ABSTRACT With the rapid growth of Web services, the demand for discovering the optimal services to satisfy the users' requirements is no longer an easy task. The critical issue in the process of service discovery is to conduct a similarity calculation. To solve such an issue, this study proposes an effective approach that combines the Embeddings from Language Models (ELMo) representation and Convolutional Neural Network (CNN) to obtain a more accurate similarity score for retrieving target Web services. More specifically, first, the study adopts the ELMo model to generate effective word representations for capturing the sufficient information from services and queries. Then, the word representations are used to compose a similarity matrix, which will be taken as the input for the CNN to learn the matching relationships. Finally, the combination of the ELMo representation and CNN is used to address the representation and interaction processes within the matching task to improve the service discovery performance. The results demonstrate the effectiveness of our proposed approach for retrieving better targeted Web services.

INDEX TERMS Service discovery, ELMo, CNN, service similarity, web service.

I. INTRODUCTION

Recent years have witnessed the continuous growth of publicly available Web services on the Internet. Additionally, the numbers and requirements of users are increased [1]. The research that focuses on retrieving the most relevant Web services to meet the specific user's requirements has received a large amount of attention [2], and it has triggered a considerable amount of effort [3], [4]. Among these studies, service discovery is becoming a critical issue in developing software applications. Generally, service discovery refers to identifying a set of candidate services from a service registry by comparing the matching degree with the given requirements of the users [5]. It can help to return the target services for users without knowing the topology deployment of the entire architecture, and it can dynamically update the service registry to ensure the real-time remote inquiries of users [6]. Furthermore, it can support businesses to better understand users and make quick improvements, to adapt their business to the environment [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Qi Zhou.

Currently, the existing work on Web service discovery mainly involves two patterns, which are syntactic discovery and semantic discovery [1]. The syntactic discovery approaches are usually developed to match the keywords of services with the user's requirements by using Information Retrieval (IR) techniques [5]. Although these approaches are easy to implement, the accuracy of the discovery outcomes suffers from several keyword-related issues [8], such as the keyword polysemy problem and the insufficient keywords provision. To overcome these issues, the semantic approaches are developed to formalize the service description and make the semantics clearer by utilizing the power of ontologies [2]. In this way, the discovery performance achieves an improvement. However, the semantic discovery approaches normally take the semantics to be the concepts and perform the term matchings without considering the relationships among the concepts and the analysis of the matching interactions [9]. This could have a serious effect on the discovery outcomes.

Indeed, conducting the service discovery requires not only an accurate description but also an effective matching mechanism between the Web services and users' requirements [5]. As addressed by Zhang *et al*. [10], discovering a satisfactory service lies in having a precise description and performing further analysis of the matching relationships. Therefore, it is important to pay more attention to the multiple stages of the matching process in service discovery. To improve the matching performance, this study investigates the existing text matching approaches. The related studies can be found in the literature. Some studies focus on the text representation (e.g., [11]), while others look at the matching interaction (e.g., [12]). Nonetheless, our study enables us to identify three major shortages.

First, the commonly used methods for learning word representations allow only a single context-independent representation for each word [13]. This circumstance implies that regardless of what the context information of new text is, the word representation will not be changed. This approach has difficulty in varying across linguistic contexts as well as modeling the complex characteristics of words, such as the syntax and semantics [14]. Second, the existing neural matching models merely focus on either representation generation or interaction matching during the process of text matching, ignoring the combination of the two stages, which could largely limit the matching performance to achieve desirable outcomes. Third, although some deep learning models, such as DSSM, have been employed in previous work to separately capture the semantic representation of each text, insufficient attention has been paid to the interaction between each representation at a lexical level [15]. It can be argued that without sufficient attention, text matching approaches still face significant challenges in achieving better service discovery. In addition to improving the text representation, it is necessary to attempt to further analyze the interactions in the word granularities.

Therefore, this study focuses on the above-mentioned issues and aims to make use of the text matching approaches in conducting precise Web service discovery. More importantly, the following research questions will be investigated in this study: (1) *What methods can obtain the word representations that capture sufficiently the syntactic and semantic information and suit the dynamic context?* (2) *How does the method learn the matching relationships between the descriptions of the service and queries at a lexical level during the service discovery process?* (3) *What refinements can be made to improve the performance for service discovery according to the specific characteristics of the neural matching models*? To answer these questions, this study develops a service discovery method that combines the Embeddings from Language Models (ELMo) representation and Convolutional Neural Network (CNN) to obtain accurate similarity scores for retrieving the optimal target Web services. The experiments are conducted across multiple evaluation metrics, and the results show that the proposed method can better capture the representations and learn the interaction relationships to improve the matching accuracy.

This study contributes to service discovery research by developing a new semantic similarity measurement that integrates the ELMo representation and CNN to enhance the service discovery. This approach will help developers to increase their understanding of the multiple stages of the matching process as well as strengthening the knowledge of optimizing the service matchmaker. Our study can also help scholars to identify shortcomings in the text matching process and to better understand the solutions for capturing the sufficient matching information for calculating the final similarity, to achieve more accurate service discovery outcomes.

The remainder of this paper is structured as follows. In Section II, we review the related work on service discovery and text marching, and we point out the importance of capturing the precise representation of texts and the rich patterns in their interactions. In Section III, we introduce the ELMo-CNN-based service discovery method used in this study. We then detail our experimental results, and we discuss the issues of special values and data fluctuation in Section IV. Finally, our conclusions, implications, limitations, and future work are described in Section V.

II. LITERATURE REVIEW

A. WEB SERVICE DISCOVERY

Due to the features of mobility, flexibility, collaboration and security, the Service Oriented Architecture (SOA) is becoming an important aspect of the Internet era [16]. Web service discovery is an important part of SOA, which has been widely investigated in recent decades [10]. With the similar aim of retrieval techniques in diverse scenarios, such as document retrieval [17], image retrieval [18] and 3D model retrieval [19], Web service discovery can be defined as a process that returns a list of relevant services to satisfy the specific requirements of service requestors [20]. Generally, the service discovery process differs by the Web services description [8]. A variety of Web service description languages and models have been proposed in both industrial and academic entities [21], going from a simple piece of plain text to large and sophisticated semantic descriptions by means of ontologies [22]. For example, a Web Service Description Language (WSDL) provides syntactic functional information on Web services with low-level message exchange descriptions [23]. WSDL allows service providers to describe a Web service with respect to what it does and how to invoke it [24]. However, according to the syntactic nature of WSDL and the heterogeneity of Web services, the inefficiencies might be acquired in exploiting Web service discovery since it can hardly find a suitable service for the service requesters by focusing only on the syntactic information [25], [26].

To overcome such shortages and enhance the interaction between humans and computers [26], the Semantic Web Service (SWS) is proposed to complement the knowledge-poor syntactic industry standards and thus to facilitate the automation of Web service-related tasks [27]. Moreover, Ontology Web Language for Service (OWL-S), as one of the well-known SWS description approaches, is built on an Ontology Web Language (OWL) that enables us to describe the properties and capabilities of Web services [28]. In addition, the Web Service Modeling Ontology (WSMO) is introduced as a conceptual framework to promote the automation of services over the Web for semantically describing all relevant aspects of SWS [29]. Both OWL-S and WSMO work on offering a concise representation of Web services with rich semantics [30]. The differences between OWL-S and WSMO mainly result from their syntax, involved parts and various reasoning methods to describe the logic behavior of a Web service [22].

According to the different service description methods, there are two types of service discovery in general, which are the syntactic-based and semantic-based approaches [8]. The syntactic-based approaches mainly discover Web services at a syntactic matching level, where the keywords of the services are matched with the user's requirement by using IR techniques [5]. For example, Hao *et al.* [31] proposed a novel IR-Style mechanism for automatically discovering and ranking Web services in response to given textual description of the desired requirements. Furthermore, to better provide the textual metadata from service description documents, Wu [32] systematically investigated the issue of WSDL term tokenization for IR-Style Web service discovery. Although these IR techniques and intelligent approaches have been incorporated to enhance the discovery performance, the syntactic service discovery approaches still suffer from low accuracy due to the heterogeneity of the Web services and the users' requirements [1].

In contrast, the semantic-based approaches aim to search for the available services, which are semantically similar to the requirements by utilizing the power of ontologies [2]. For example, Lu *et al.* [33] developed a multiple semantic fusion (MSF) model to generate the sense-specific vector and performed the MSF model-based similarity measure for semantic-oriented service discovery. Zhang *et al.* [34] studied service goals extraction and user query expansion through depth analysis of the linguistic information of various texts. The results were to determine an innovative solution to accelerate the improvement of the discovery performance. Thus, the semantic methods can better identify the similarities and improve the performance of matchmaking [35], [36].

Currently, a number of attempts have been made to focus on solving the issue of properly discovering the required services within the semantic-based approaches (e.g., [5], [37]). Among these efforts, the similarity-based approaches are very representative [5]. These approaches measure the relevance degree between the Web service and the user's requirements by performing a similarity calculation. For example, Chen *et al.* [2] developed a measurement of semantic similarity to enable more accurate service-request comparisons. This measurement addresses the multiple conceptual relationships and combines the interface signature with a textual description for better Web service discovery. However, Chen *et al.* [8] argued that a semantic similarity measurement must consider both the outside functionality and inside processes. This contention is further supported by Hu *et al.* [38], who is concerned with the integration of parameters and operations during the process of similarity computing. The above-mentioned studies conduct the similarity calculation from the different views and perform the integration to represent the final similarity. However, in consideration of the functional capability of a Web service, the textual description provides more specific information than the interface signatures or other parameters [21]. Therefore, it is important to adopt text-based service discovery [39] and focus on the development of an efficient text matching approach to achieve better discovery performance.

B. TEXT MATCHING

Natural Language Processing (NLP) is an important field for realizing effective communication between humans and computers [40]. Among the various tasks in NLP, text matching is a fundamental task, and it has been used in a wide range of applications, such as information retrieval, question-answering, and machine translation [12]. The process of text matching can be defined as the method that returns the matching scores for the compared texts by performing a posterior similarity measurement [41]. Existing approaches of text matching usually involve two stages, which consist of generating the text representation and calculating the matching scores. The well-known matching model is Cosine TF$\times$IDF [42], which generates Cosine similarity scores based on the TF$\times$IDF vectors of texts. Similarly, BM25 is conducted through calculating the sum of the term IDFs multiplied by the respective normalized term frequencies [43]. Although these methods are simple to implement, the issues of mismatching are inevitable due to the inherent defects within these methods.

There is an urgent need to improve the matching process at every stage. The understanding of the similarity between texts is not a simple task for machines because both the syntactic and semantic information must be considered [44]. Poria *et al.* [45] integrated common-sense computing to calculate the word distributions in the Latent Dirichlet Allocation (LDA) algorithm, which enables the shift from syntax to semantics. Similarly, Zhu *et al.* [46] proposed a semantic similarity-based method that aims at improving the purity of the LDA results. Moreover, Yoshua *et al.* [47] indicated that distributed word embedding is usually constructed in an unsupervised manner from large unstructured corpora. It can explicitly encode many linguistic regularities and can be utilized to extract the semantic features of texts. Furthermore, several studies that focus on pretrained embedding provide insights into generating word representations with sufficient information. For example, Nguyen *et al.* [48] presented the interdependent representation of short text pairs based on lexical semantics by using pretrained word embeddings and external sources of knowledge. The results show that the interdependent representations of short text pairs are effective and efficient for semantic textual similarity tasks. Additionally, Mikolov *et al.* [11] proposed two novel model architectures for computing continuous vector representations of words from large datasets. Pennington *et al.* [49] performed training on aggregated global word-word co-occurrence statistics and used an unsupervised learning algorithm to obtain the vector representations for words.

Although prior studies point out various methods to promote the matching process, they still have difficulty in identifying the different semantics of words within different contexts and resolving the problem of the combinatorial structure of language. For the similar issues found in other areas, many researchers have paid attention to alleviating the effects of feature modeling on task performances. For example, Liu *et al*. [19] proposed an original clique-based graph matching method that considers both multimodal and multiview information for 3D model retrieval. Similarly, to achieve better performance for 3D object retrieval, Gao *et al*. [50] developed a new method called Multiview Discrimination and Pairwise CNN. This method can be used to realize the simultaneous input of multiple batches and multiple views and obtain the features that have better intraclass compactness and interclass separability. In addition, Gao *et al*. [51] developed a nature-inspired adaptive fusion and category-level dictionary learning model for multiview human action recognition, where the latent complementary relationships among different views are modeled by using convex hull coefficients. Evidence from these studies points out that capturing the sufficient word information and generating the precise word representations to suit the dynamic context can make great contributions toward accomplishing the text matching task.

Furthermore, the similarity measurement between the texts has direct impact on the matching accuracy. However, most of the previous similarity measurements are simply performed based on the computational vector metric [52]. The similarity score is determined by using the corresponding functions, such as Cosine similarity, Euclidean distance, or Manhattan distance [53]. However, it provides a limited degree of semantic similarity and fails to suit all types of input data. To overcome this issue, a variety of studies focus on integrating the similarity metrics to receive a more accurate similarity score. For example, Alves *et al*. [54] combined the lexical, syntactic, semantic, and distributional similarity through a regression function. Kiela and Clark [55] studied the effects of different computational metrics on semantic similarity estimation, dimensionality reduction strategy and the feature granularity of vectors. Based on these efforts, Lu *et al*. [52] introduced a differential evolution-based approach by aggregating the corpus-based with the WordNet-based similarity metrics to make significant improvements. Although these previous methods have made remarkable progress, they still lack further analysis of the interaction process and the exploration of interaction features.

In view of the fact that the neural network has great performance and in addition requires a lower level of manual operations, diverse neural network architectures have been applied to complete the tasks in NLP. For example, Minaee *et al*. [56] presented a comprehensive review of different deep learning-based models for text classification tasks, such as sentiment analysis, news categorization, and topic classification. Additionally, to provide a framework with great scalability and reliability, Zhao *et al*. [57] built a unified capsule network (NLP-Capsule) and demonstrated its effectiveness on multilabel text classification and question answering tasks. With regard to neural-based text matching, much attention has been drawn to solving the problems in the stages of text representation generation and matching score calculation [58].

Aligned with the aims of providing the capacity of extracting important text features and learning complex functions, the representation-based and interaction-based neural matching models are developed while varying on each stage [59]. The representation-based models (also known as distributed models) normally work independently to learn a representation for each text and calculate the similarity between the two estimated representations with simple vector functions. For example, DSSM [60] combined a feed-forward neural network and cosine similarity to match a given query string and document title. Shen *et al*. [61] improved the representations that are generated by using the CNN. Architecture-I (ARC-I) [12] took the convolutional approach to make the representations and performed the matchmaking with a multilayer perceptron. At the same time, the interaction-based models use simple representation mapping functions, in which a neural model computes detailed interactions and matching features between the compared texts. For example, DeepMatch [62] mapped each text to a sequence of terms and trained a feed-forward network that is powered by a topic model over a word interaction matrix to compute the matching score. In addition, MatchPyramid [12] and Architecture-II (ARC-II) [12] applied a deep neural network for constructing the interaction matrix between the word embeddings of texts.

However, it is worthwhile to note that both the representation-based and interaction-based neural models paid attention to only a single stage of the matching process. A successful text matching algorithm must capture not only the precise representation of texts but also the rich patterns in their interactions [12]. As addressed by Gao *et al*. [50], it is necessary to construct an end-to-end network by integrating the feature extraction and object retrieval processes to learn the similarity measurements from the data. Therefore, our study develops a method that combines the ELMo representation and CNN approaches to achieve more accurate service discovery. In doing so, the differences in the text representation under a dynamic context can be recognized, and the useful interaction features can be captured to improve the matching performance.

III. RESEARCH METHOD

This section details our ELMo-CNN-based method for service discovery. Fig. 1 shows the overall framework of our proposed method, which is composed of three main parts. The first part is the data preprocessing, which is used for supporting the research implementation by detailing the service or query descriptions. The second part is about the generation process of the ELMo representation, which aims to obtain the effective word representation by performing the

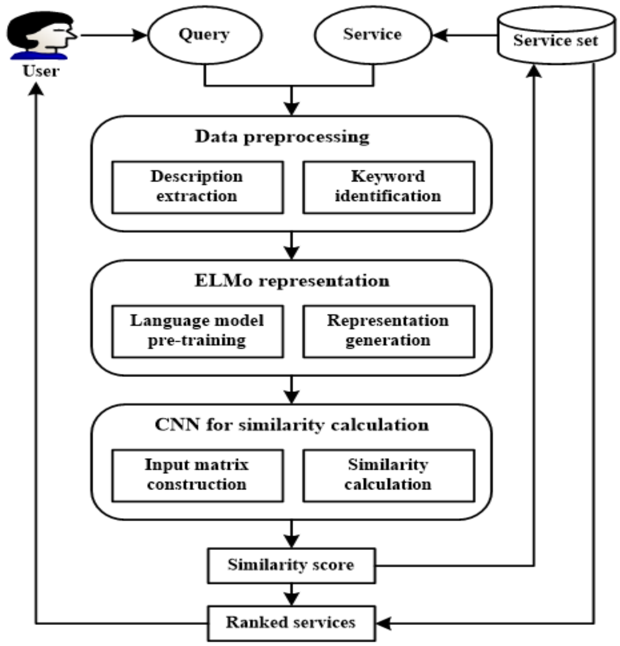

FIGURE 1. The framework of the proposed method.

pretreatments and adopting the pretrained ELMo model for each word. The third part focuses on the similarity calculation through the CNN, which uses the word representations as input and conducts the convolution and pooling operations to achieve the optimal ranked services.

Please note that it is essential to clarify the relations of the ELMo and CNN in forming the service discovery method in this study. Although each part of the method plays an important role and has its own specific characteristics and functions, the ELMo and CNN works as a whole for achieving the efficiency of the service discovery in this study. In other words, the ELMo and CNN are closely interrelated, and thus, we cannot separately compare their importance in this study.

A. DATA PREPROCESSING

Fig. 2 shows an example to illustrate the storage of a specific service in an OWL-S document. As shown in the figure, the detailed service description is highlighted in red. As the proposed discovery mechanism is on the basis of a textual description, the corresponding descriptions of the service and query are manually extracted from the profile module, which indicates the equipped or required functionality to support the matching process.

A total of three test queries were used in our study, a number that is considered to be acceptable by other scholars (e.g., [10], [63]). This finding is further supported by Zhang *et al.* [10], who argued that using three queries that pertain to different domains is useful to obtaining comparative results, and it can reduce the complexity at the operational level. A description of these test queries is shown in Table 1.

Prior to generating the ELMo representation for queries and services, data preprocessing was performed first.


```
<owl:Ontology rdf:about="">
<owl:imports rdf:resource="http://127.0.0.1/ontology/Service.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/Process.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/Profile.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/Grounding.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/concept.owl" />
</owl:Ontology>

<service:Service rdf:ID="_PRICE_SERVICE">
<service:presents rdf:resource="#_PRICE_PROFILE"/>
<service:describedBy rdf:resource="#_PRICE_PROCESS"/>
<service:supports rdf:resource="#_PRICE_GROUNDING"/>
</service:Service>

<profile:Profile rdf:ID="_PRICE_PROFILE">
<service:isPresentedBy rdf:resource="#_PRICE_SERVICE"/>
<profile:serviceName xml:lang="en">
Three wheeled Car price service
</profile:serviceName>
<profile:textDescription xml:lang="en">
This service provides the price of a 3(three) wheeled car with Audi brand.
</profile:textDescription>
<profile:hasOutput rdf:resource="#_PRICE"/>

<profile:has_process rdf:resource="_PRICE_PROCESS" /></profile:Profile>

<!--<process:ProcessModel rdf:ID="_PRICE_PROCESS_MODEL">
<service:describes rdf:resource="#_PRICE_SERVICE"/>
<process:hasProcess rdf:resource="#_PRICE_PROCESS"/>
</process:ProcessModel>-->
```


FIGURE 2. The owls information of a specific service.

TABLE 1. The description of given queries.

Query	Description
Q1	This service returns author of the given science-fiction book.
Q2	This is a service that provides the geographic location.
Q3	This service gives you information for the scholarships offered by the given government.

It includes removing stop words, splitting and keyword identification with the purpose of obtaining the useful words from the extracted descriptions.

TABLE 2. An example for comparing the semantics of the same word.

S1: I read the book last week?
S2: Can you read the letter now?

B. ELMo REPRESENTATION

To illustrate the characteristics of the ELMo representation, an example of comparing the semantics of the same word is presented in Table 2. It is clear that both sentences (S1 and S2) have the common word of "read". However, the meaning of "read" in each sentence is fairly different. Moreover, the tense of "read" in S1 is preterit, whereas in S2, it represents the present tense. Traditional word embedding methods could create the same representations for "read" since the traditional methods hardly identify the context and have difficulty in distinguishing polysemous words. Unlike the traditional methods, the word embeddings derived from ELMo are in line with the context of different sentences and are computed based on the learned functions of all of the internal layers in the bidirectional Long Short-Term Memory (LSTM). In consequence, the representations of the "read" in this example differ from each other due to their context of use. The ELMo representation shows great advantages from the perspective of generating contextualized representations. Therefore, it is vital to use the ELMo representation to a broad range of NLP tasks, especially for completing the text similarity calculation.

In our ELMo-CNN-based method, the similarity of descriptions derived from each service and query are addressed, which determines the ranking scores of candidate services and has a significant impact on the final retrieved accuracy. For the process of a description similarity calculation, it is of great importance to conduct feature mapping from natural text to the corresponding word representations that capture the plentiful syntactic and semantic information. Furthermore, to consider the issues of context regardless and to dynamically update the word embeddings, the ELMo representation is thus employed after the data processing in this study.

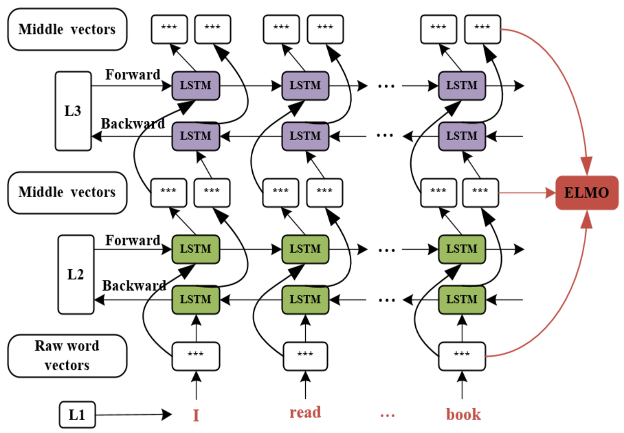

FIGURE 3. **The architecture of the pre-trained ELMo representation model.**

1) MODEL ARCHITECTURE

Fig. 3 presents the architecture of the ELMo representation model that is constructed by three layers, namely, L1, L2, and L3. L1 receives the original text and extracts the raw word vectors that are going to be used as the inputs for the contextualized learning within L2 and L3. It can be seen that L2 and L3 adopt the bidirectional LSTM, including forward LSTM and backward LSTM. Each layer of the 2-layer LSTM iteratively implements the bidirectional operations, which can provide abundant contextualized information for the intermediate vectors. The corresponding output vector can be obtained from each layer. Finally, the linear combination of the raw vector generated from L1 and the two intermediate vectors from L2 and L3 are taken as the expected ELMo representation for each word.

The process of obtaining the ELMo representation involves two stages. The first stage is to pretrain a language model, and the second stage is about the embeddings generated by using the pretrained model. To better understand the mechanism of the ELMo representation, it is necessary to describe the language model first.

2) LANGUAGE MODEL

The language model is a simple, unified, and abstract formal system, which has been widely used in the fields of NLP, such as speech recognition, machine translation, Chinese automatic word segmentation and parsing [64]. With the purpose of calculating the probability of a sentence, the language model is therefore used in our study. Assume that given a sentence with n tokens, $W = (w_1, w_2, \ldots, w_n)$, where w_i is the *ith* token in this sentence. The probability of W appearing in this order can be expressed as follows:

$$\begin{aligned} p(W) &= p(w_1, w_2, \ldots, w_n) \\ &= p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|w_1^2) \\ &\quad \cdot p(w_4|w_1^3) \cdot \ldots \cdot p(w_n|w_1^{n-1}) \end{aligned} \tag{1}$$

In (1), $p(w_n|w_1^{n-1}) = p(w_n|w_1, w_2, \ldots, w_{n-1})$, and $p(w_n|w_1^{n-1})$ is the parameter of the language model.

3) PRETRAINING PROCESS

The basic component embedded in this structure is LSTM, which makes a model that has great functionality in terms of its memory. This finding means that the next word will be predicted when the previous words have been known or remembered. Based on this functionality, the ELMo representation model adopts the bidirectional LSTM to learn the context relations among the sequence of words. To be specific, given a sentence with n tokens, $W = (w_1, w_2, \ldots, w_n)$, and the forward LSTM calculates the probability of the sentence by modeling the probability of *ith* token w_i in this sentence on the basis of the given sequence $(w_1, w_2, \ldots, w_{i-1})$, as shown in (2).

$$p(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} p(w_i|w_1, w_2, \ldots, w_{i-1}) \tag{2}$$

Similar to the forward language model, the backward LSTM can be conducted in an analogous way to predict the previous token based on the future context as follow:

$$p(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} p(w_i|w_{i+1}, w_{i+2}, \ldots, w_n \tag{3}$$

Both of the forward and backward models constitute a bidirectional language model, and the pretraining process aims to maximize the log likelihood of these two directions.

$$\begin{aligned} \sum_{i=1}^{n} (&\log p(w_i|w_1, w_2, \ldots, w_{i-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) \\ &+ \log p(w_i|w_{i+1}, w_{i+2}, \ldots, w_n; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s)) \end{aligned} \tag{4}$$

In (4), Θ_x and Θ_s stand for the parameters for context-independent token representation and the soft-max layer, respectively, both of which can be shared in forward and backward LSTM. Additionally, $\overrightarrow{\Theta}_{LSTM}$ and $\overleftarrow{\Theta}_{LSTM}$ are the separate parameters for each directional LSTM.

4) GENERATION OF WORD REPRESENTATIONS

Since the descriptions of the services and queries are represented as natural text, it is necessary to generate the corresponding numerical vectors that can fully capture the key features. Having constructed the pretrained bidirectional language model in the corpus, three embeddings of each token can be obtained for the given text. These embeddings are derived from each layer of the ELMo representation model. Among them, one is the basic raw vector, and others are the intermediate vectors from the 2-layer bidirectional LSTM. As indicated by Matthew *et al.* [14], the higher layer LSTM states capture the context-dependent aspects of the word meanings, which are beneficial for extracting the semantic information. At the same time, the lower layer LSTM can capture more syntax information. Additionally, because we conduct the similarity calculation at a lexical level, the two-dimensional matrix that consists of three embeddings with 1024 dimensions is extracted from the three-dimensional tensor for each constituted keyword. We note that most of the NLP tasks are conducted on the basis of one-dimensional vectors. Therefore, the superposition of the three embeddings stacked above each keyword is used as the final ELMo representation to complete the similarity calculation.

C. CNN FOR THE SIMILARITY CALCULATION

This study primarily finds an efficient discovery method to retrieve the optimal service that conforms to the input query. During the process of discovery, it is essential to conduct an effective description similarity calculation. Therefore, the ELMo representation and CNN are combined with the purposes of capturing the useful word representations as well as learning the interrelations among pairs of words. Fig. 4 shows the specific process of the similarity calculation, and it details the description of how to select the suitable services on the basis of similarity scores.

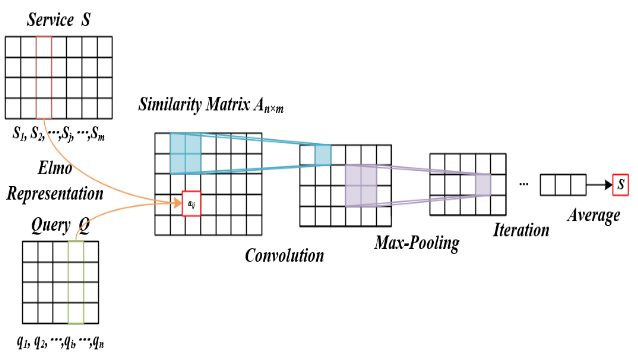

FIGURE 4. The process of similarity calculation.

1) INPUT MATRIX CONSTRUCTION

It is critical to receive accurate similarity by implementing the CNN within our ELMo-CNN-based discovery method. The keywords and their corresponding ELMo representations of services and queries are identified after the ELMo processing. This approach provides a solid foundation for the similarity calculation. In the following paragraphs, the input matrix construction for the CNN is described.

The input of the CNN is the similarity matrix between the service and query in the word granularity. More specifically, once a query appears, it must be matched with each candidate service to determine the final target services. For each of the pairs of queries and services, it assumes that the given query is with n key words, $Q = (q_1, q_2, \ldots, q_n)$, and the number of these within a service is m, $S = (s_1, s_2, \ldots, s_m)$. The Cosine similarity of the pairs of keywords in a service and query is used to construct the $n\times m$ dimensional input matrix based on their ELMo representation.

TABLE 3. The similarity matrix in word granularity.

Service \ Query	title	discover	film	matching	given	using	service
service	0.6246	0.5996	0.6273	0.6028	0.5862	0.5833	**0.8551**
gives	0.5949	0.6643	0.5847	0.5905	0.6806	0.6053	0.5878
information	0.6177	0.6325	0.6817	0.6281	0.6167	0.5959	0.6681
scholarships	0.6256	0.6040	0.6559	0.6353	0.6050	0.5640	0.6184
offered	0.5878	0.6026	0.5937	0.5883	0.7585	0.6192	0.6100
given	0.6078	0.6254	0.6156	0.6255	**0.8755**	0.6463	0.6269
government	0.5979	0.5861	0.5930	0.5779	0.5983	0.5934	0.6660

Table 3 shows an example of the similarity matrix for illustrating the construction process of the CNN input matrix. The query and a random service are set as "*The service gives you information for the scholarships offered by the given government*." and "*One can discover a film title that is matching given title by using this service*.", respectively. During the process of ELMo representation generation, the query and service descriptions have been split into several keywords (see the bold titles in Table 3). For example, with regard to the query, the keywords of "title", "discover", "film", "matching", "given", "using", and "service" are extracted to represent the query. Each cell of the table is the value of the cosine similarity between the corresponding pair of keywords from the query and service. Note that there are two common words in the keywords set, namely, "given" and "service". However, the corresponding similarity values are not 1.0000 (highlighted in bold), which differs from the conventional understanding that the similarity value between the same words is supposed to be 1.0000. This concern could arise because the word embeddings are generated based on their own context. Differences in the representations of the same words could exist.

2) SIMILARITY CALCULATION

Since the input of the CNN is a two-dimensional similarity matrix in this study, we mainly focus on the convolution layer and pooling layer within the hidden layers to form the core modules that would be used to realize the feature extractions. Differing from the CNN used in other fields, such as image recognition, the size of the convolution filter is set to 2*2, and only one filter is adopted to easily capture the similarity features.

In the process of the similarity calculation, the obtained similarity matrix is performed as the input for the subsequent operations. First, we extract the primary features in the convolution layer by scanning the input matrix with a convolution filter that focuses on the local feature extraction and determines the receptive field in the convolution operation. The element in the output feature matrix is taken as the sum of the product from each element in the input feature matrix and the corresponding element in the convolution filter. At the same time, we scan the complete input matrix with one stride to construct the output feature map in the convolution layer.

Similarly, the output map is delivered as input to the next pooling layer. The Max-Pooling strategy is adopted in this study, which can keep the max value in the receptive field without the redundant features. Since the dimension of the feature matrix can be decreased after the convolution and pooling operations, it is necessary to identify the dimension of the current feature matrix to determine whether to perform the pooling operation. If the matrix had decreased to a one-dimensional vector, we would use the average of the elements within the vector as the final similarity score. Otherwise, the convolution and pooling operations would be repeated until the one-dimensional matrix could be achieved. For each query, the similarity score of each service in the dataset can be obtained by the ELMo-CNN-based discovery method. The service that receives the highest score is considered to be the target service that could better match with the given query.

IV. EXPERIMENTAL WORK

This section provides the details of our experimental work to validate the proposed ELMo-CNN-based discovery method. This section covers the dataset, evaluation metrics, results, and corresponding discussion.

A. DATASET

The OWLS-TC4 is a publicly available test collection that is used for service retrieval, which is composed of a number of Web services and queries across a wide range of application domains, such as communication, economic, and education, among others. Having a close look at the profile of each service and query, some services and queries with similar descriptions were found. To minimize the effects of our results and to reduce the operational complexity, 430 services and 3 queries were selected from the initial collection to form the experimental dataset. Both the services and queries are represented in OWL-S profile documents (please see Fig. 2). Furthermore, a set of graded relevance was predefined on a scale of 0-3 in the dataset ("3" denotes that a Web service is highly relevant to a given query, "2" means a relevant relation, "1" indicates potential relevance, and "0" shows an irrelevant relation).

B. EVALUATION METRICS

To conduct a comprehensive evaluation and receive the consistent results, the performance of our ELMo-CNN-based discovery method was evaluated in terms of the accuracy and error analysis. The detailed descriptions of the evaluation metrics are provided in the following subsections.

1) METRICS FOR ACCURACY

Three commonly used metrics, namely, P@k (Precision), AP@k (Average Precision) and NDCG@k (Normalized Discounted Cumulative Gain), were employed to assess the top k recommended services for accuracy. The values of k were set at 5, 10, 15, and 20 in our study.

P@k is one of the most intuitive metrics in the field of search engines. It normally refers to the fraction of relevant services in the first k retrieved Web services for a given query. As shown in (5), m_k represents the number of relevant services within the top k retrieved services.

$$P@k = \frac{m_k}{k} \tag{5}$$

With regard to the order among the ranked services, P@k is insufficient to reflect the performance since it only considers the number of relevant retrieved services. Therefore, AP@k and NDCG@k were involved in the evaluation according to the relevance of each position in the ranked service list. In (6), N_{rel} denotes the number of relevant retrieved services; $P(i)$ represents the precision of the top i services; and $rel(i)$ indicates the relevance of the service at the ranking position i to the given query, which is the same as rel_i in (7). Moreover, IDCGk is the maximum possible DCG until position k by sorting the retrieved services in terms of the relevance. The final value of NDCG@k is calculated by using (8).

$$AP@_k = \frac{\sum_{i=1}^{k}(P(i)^* rel(i))}{N_{rel}} \tag{6}$$

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{\log_2(i+1)} \tag{7}$$

$$NDCG@k = \frac{DCG_k}{IDCG_k} \tag{8}$$

As mentioned earlier, each service was predefined on a scale of 0-3 during the process of the evaluation. In contrast to NDCG, which allows the relevance to be the real value, AP uses a binary value, where 1 indicates relevance and 0 indicates irrelevance, to implement the evaluation. Thus, the relevance values that are greater than or equal to 2 were set to 1, while the others were set to 0, when we conducted the P and AP evaluations.

2) METRICS FOR ERROR

In addition to performing the accuracy evaluation, the MAE (Mean Absolute Error) and RMSE (Root Mean Square Error) were used to assess the errors. The error analysis is calculated on the basis of differences between the prediction ($\hat{y}_i$) and actual ratings (y_i) for each service (please see (9) and (10)). In (9) and (10), m refers to the number of retrieved services in a ranking list.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|(\hat{y}_i - y_i)\right| \tag{9}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \tag{10}$$

C. CONTRASTIVE EXPERIMENTAL DESIGN

To retrieve the most relevant services by better measuring the matching degree to a given query, the comparative approaches were used to validate the effectiveness of our proposed ELMo-CNN-based method. The comparative approaches cover both representation-based and interaction-based perspectives.

The representation-based methods aim to improve the discovery performance only through providing effective text representation. The representation of the services and queries were generated by using the LDA [65], Doc2vec [66], and CLSM (Convolutional Latent Semantic Model) [67] models. Afterward, the corresponding cosine similarity was calculated between each service and query. In contrast, the interaction-based method pays more attention to the process of similarity calculation. It used the Word2vec model [68] to obtain the word representations of the services and queries. These provide the input of the CNN to implement the similarity calculation. Accordingly, the comparative approaches are denoted as LDA, Doc2vec, CLSM, and W2v-CNN. However, it is worthwhile to note that both types of comparative approaches merely focus on improving the performance in the stages of representation generation or similarity calculation, which shows the insufficiency in obtaining the accurate similarity for the complete text matching procedure. To overcome this shortcoming, the ELMo-CNN-based method, therefore, was employed in this study. In doing so, it can not only obtain more effective representations but also conduct more efficient similarity calculations to discover the desirable target services for meeting the user's requirement.

D. RESULTS AND DISCUSSION

For each query, we performed each method to retrieve twenty ranked services that are used for the evaluation. To indicate the differences in the performance between our proposed method and the comparative approaches, the description of the top 5 ranked services are matched with the given queries.

Table 4 shows an example of comparing the top 5 ranked services for Q1. As shown in the table, the specific service descriptions are presented based on the ranking positions in the service recommendation list through different discovery methods. The smaller ranking number indicates a greater correlation. Moreover, the words within the service descriptions, such as ''author'' and ''science-fiction'' (see the words in bold), have been found and used to represent the key information of the query, which determines the relevance level of the retrieved services.

TABLE 4. The comparison of the ranked top 5 services for Q1.

Methods	Ranking	Top 5 services for Q1
LDA	1	This service returns **author** of the given **science-fiction** book.
	2	This service returns **author** of the given **science-fiction** novel.
	3	This service informs about drought in a given municipal-unit.
	4	Merchant service returns prepared food of the certain store.
	5	This service returns farmlands which provide given sports activities.
Doc2vec	1	This service informs you for a person name who works as **co-publisher** of a certain given book title.
	2	This service returns lecturer who works in a given learning center organization.
	3	This service informs which type of investigating facility is available in a given care organization.
	4	This service places the book ordered for purchase by an authorized person in his shopping cart.
	5	This service returns name of **publisher** of a given certain book title.
CLSM	1	This service returns **author** of the given **science-fiction** book.
	2	This service returns **author** and recommended price of a given **science fiction** book title.
	3	This service returns **author** of the given **science-fiction** novel.
	4	This service returns list of current purchase prices of a given book title.
	5	This service returns US tax on the given prepared food.
W2v-CNN	1	Moves the agent from given current position to calculated future position.
	2	This service returns **author** of the given **science-fiction** book.
	3	This service returns **author** of the given **science-fiction** novel.
	4	This service returns **author** or authors of a given **science fiction** novel.
	5	This service returns **author** of a certain short-story.
ELMo-CNN	1	This service returns **author** of the given **science-fiction** book.
	2	This service returns **author** of the given **science-fiction** novel.
	3	This service returns **author** or authors of a given **science fiction** novel.
	4	This service returns **author** and recommended price of a given **science fiction** book title.
	5	This service returns **author** of a certain short-story.

Moreover, it can be found that the results of Doc2vec show the lowest accuracy in comparison with the other approaches. More specifically, among the top 5 retrieved services by using Doc2vec, only two services are close to Q1. Additionally, although the keywords of the two services, namely, ''co-publisher'' and ''publisher'', have the connection with ''author'', the major word, namely, ''science-fiction'' is hardly identified. It is interesting to see that the two relevant services are ranked as No. 1 and No. 5, respectively, while the other three services in the middle have no relation with the given query. Therefore, the findings from results show that the discovery accuracy of Doc2vec has been influenced.

In contrast, the results of LDA, CLSM, and W2v-CNN have significant improvements. With regard to LDA and CLSM, the top 2 and top 3 services are closely related to the query, and the services ranked first are exactly matched with Q1. Such results suggest that the ranking positions of the retrieved services are consistent with the relevance level, in contrast to the results from Doc2vec. In addition, the keywords of the relevant services are better captured through LDA and CLSM, which are the same as Q1. However, the common drawback in the results of LDA and CLSM is that not all of the five services conform to the query. For example, the last three services of LDA and the last two of CLSM

TABLE 5. The comparison of the ranked top 5services for Q2.

Methods	Ranking	Top 5 services for Q2
LDA	1	This is a service that provides the **geographic location**.
	2	This service returns accommodations info of the given geopolitical-entity.
	3	Closes the given open and unlocked door of a room.
	4	A magical price of a red Ferrari car F50 model.
	5	This service allows to create a static map **geographic locations**.
Doc2vec	1	SR service returns available camera and its recommended price in the certain shopping mall.
	2	This service informs the latest price of certain type of apple available in the apple market.
	3	This service returns information of available hotels in a given city of Germany.
	4	This service informs you about the prepared food which is not available in the certain grocery store and its required quantity.
	5	This service informs you about available digital SLR, its price and availability time in the certain shopping mall.
CLSM	1	This is a service that provides the **geographic location**.
	2	This service returns a recommended price of a set of MP3 Player and a DVD Player.
	3	This service provides a cola for the maximum price and quality.
	4	This service informs the latest price of certain type of apple available in the apple market.
	5	This service returns current weather front of Germany.
W2v-CNN	1	This is a service that provides the **geographic location**.
	2	MerkelD is a German firm and presents a service.
	3	Returns a publication using the academic item number.
	4	This service provides information about funding offered by a given government for a given award.
	5	This service provides you information about lending offered by a given government for a given academic degree.
ELMo-CNN	1	This is a service that provides the **geographic location**.
	2	This service provides information bearing objects.
	3	This service allows to create a static map **geographic locations**.
	4	This service handles physician authorizations.
	5	This service returns destinations which provide given activity.

TABLE 6. The comparison of the ranked top 5 services for Q3.

Methods	Ranking	Top 5 services for Q3
LDA	1	This service gives you information for the **scholarships** offered by the given **government**.
	2	The service returns a list of flights that fulfill the desired travel conditions.
	3	This service informs about lending offered for the academic-degree by the German **government**.
	4	This service informs about funding offered for the academic-degree by the German **government**.
	5	This service informs about funding offered for the award by the German **government**.
Doc2vec	1	This is a nice service to know about the cities which provide both sports hiking and surfing activities.
	2	This is the best service to know about the names of beaches which provide good surfing activities.
	3	This service returns destinations which provide given activity.
	4	This service gives you information for the lending offered by the given national **government**.
	5	This is the best service to know about good destinations for surfing.
CLSM	1	This service gives you information for the **scholarships** offered by the given **government**.
	2	It is an attractive service to know about the **scholarships** offered for a given academic-degree by the German **government**.
	3	This service gives you information for the lending offered by the given national **government**.
	4	This service informs the latest market purchase price of a given honey type.
	5	This service informs the recommended price of a three wheeled car model given the model year production.
W2v-CNN	1	This service gives you information for the **scholarships** offered by the given **government**.
	2	This service returns not only information of new comedy films.
	3	This service returns information of all famous hotels in the world.
	4	This service gives you information for the lending offered by the given national **government**.
	5	This service informs about funding offered for the award by the German **government**.
ELMo-CNN	1	This service gives you information for the **scholarships** offered by the given **government**.
	2	This service gives you information for the lending offered by the given national **government**.
	3	This service informs about funding offered for the award by the German **government**.
	4	It is an attractive service to know about the **scholarships** offered for a given academic-degree by the German **government**.
	5	This service informs about lending offered for the award by the German **government**.

show no association with Q1. For W2v-CNN, there are four relevant services within the top 5 services, although the first ranked service is irrelevant. This finding could influence the overall discovery performance even though the proportion of the relevant services has increased.

Compared with these approaches, our proposed ELMo-CNN-based method shows better performance. The results show that the retrieved top 5 services are more in line with the given query and the keywords are accurately recognized. More importantly, the ranking positions of these services are consistent with the relevance level to Q1, which implies that the services with a higher level of relevance are ranked top in the recommendation list.

Additionally, Tables 5 and 6 show the comparison results of the ranked top 5 services for Q2 and Q3, respectively. Similarly, the results show that the proposed method has better performance than the other four comparative approaches. Based on these findings, we believe that the performance of our ELMo-CNN-based method benefits from simultaneous improvement of the performance of representation generation and similarity calculation. This aspect can help to better analyze the keywords of the services and queries as well as learn their interrelationships.

Furthermore, the comparison of all of the methods under the same evaluation metric and the comparison of each method across multiple metrics are shown in this study. Tables 7, 8, and 9 present the values of P, AP, and NDCG of the top 5, 10, 15, and 20 service recommendation lists generated by using the different methods for each query. To better understand these results, we take Q1 as an example to specifically explain the results through analyzing the relationships with the corresponding service descriptions (see Tables 7 and 4).

As shown in Table 7, the values of our ELMo-CNN-based method are greater than other comparative approaches within the P value. In particular, the result of P@5 achieves the

TABLE 7. The comparison of P, AP, and NDCG for Q1.

Methods	P				AP				NDCG			
	@5	@10	@15	@20	@5	@10	@15	@20	@5	@10	@15	@20
LDA	0.4000	0.3000	0.2000	0.1500	**1.0000**	0.7778	0.7778	0.7778	**1.0000**	0.9131	0.9131	0.9131
Doc2v	0.4000	0.2000	0.1333	0.1000	0.7000	0.7000	0.7000	0.7000	0.7153	0.7022	0.6995	0.7061
CLSM	0.6000	0.3000	0.2000	0.1500	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9762	0.9762	0.9762
W2v-CNN	0.8000	0.6000	0.4000	0.3500	0.6792	0.6569	0.6569	0.6157	0.8157	0.8157	0.7985	0.7849
ELMo-CNN	**1.0000**	**1.0000**	**0.7333**	**0.6500**	**1.0000**	**1.0000**	0.9758	0.9389	**1.0000**	**0.9908**	**0.9779**	**0.9679**

Note: The last row shows the results of the proposed ELMo-CNN based method for Q1; The value in bold within each column illustrates the maximum under the specific metric.

TABLE 8. The comparison of P, AP, and NDCG for Q2.

Methods	P				AP				NDCG			
	@5	@10	@15	@20	@5	@10	@15	@20	@5	@10	@15	@20
LDA	**0.4000**	0.2000	0.1333	0.1000	0.7000	0.7000	0.7000	0.7000	0.7980	0.7843	0.7765	0.7765
Doc2v	0.0000	0.1000	0.0667	0.0500	0.0000	0.1000	0.1000	0.1000	0.6309	0.4109	0.4109	0.4254
CLSM	0.2000	0.2000	0.2000	**0.1500**	**1.0000**	0.6428	0.5194	0.5194	**1.0000**	0.7184	0.6784	0.6749
W2v-CNN	0.2000	0.1000	0.1000	0.0500	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
ELMo-CNN	**0.4000**	**0.2000**	**0.1333**	0.1000	0.8333	0.8333	0.8333	0.8333	0.8028	0.7949	0.7917	0.7985

Note: The last row shows the results of the proposed ELMo-CNN based method for Q2; The value in bold within each column illustrates the maximum under the specific metric.

TABLE 9. The comparison of P, AP, and NDCG for Q3.

Methods	P				AP				NDCG			
	@5	@10	@15	@20	@5	@10	@15	@20	@5	@10	@15	@20
LDA	0.8000	0.5000	0.4000	0.3000	0.8042	0.8100	0.7660	0.7660	0.8432	0.8490	0.8402	0.8402
Doc2v	0.2000	0.4000	0.4000	0.3500	0.2500	0.3388	0.3683	0.3657	0.5000	0.4636	0.4860	0.4976
CLSM	0.6000	0.4000	0.3333	0.3000	**1.0000**	**1.0000**	0.8714	0.7788	**1.0000**	**1.0000**	**0.9655**	**0.9393**
W2v-CNN	0.6000	0.7000	0.6000	0.5000	0.7000	0.6662	0.6579	0.6477	0.7763	0.7226	0.7328	0.7357
ELMo-CNN	**1.0000**	**0.8000**	**0.7333**	**0.6500**	**1.0000**	0.9472	**0.8834**	**0.8640**	0.9282	0.9192	0.9141	0.9156

Note: The last row shows the results of the proposed ELMo-CNN based method for Q3; The value in bold within each column illustrates the maximum under the specific metric.

highest score of 1.0000, which is consistent with the results in Table 4, thus showing that the five retrieved services and their identified keywords, "author" and "science-fiction", are highly related to the query. However, the values of P@15 and P@20 are reduced to 0.7333 and 0.6500, respectively. This finding could have occurred because with the expansion of the retrieval range, the P values can be decreased since the number of the retrieved services and the number of relevant services could hardly rise with the same ratio. Surprisingly, both LDA and Doc2v have two relevant services within the top 5 services, but the ranking positions of these services differ from each other. Despite the fact that the discovery results of LDA are intuitively better than Doc2v, the P@5 values of these two methods have no difference (0.4000). This finding could occur because the Precision focuses on the number of relevant services without considering the ranking position of each service.

To provide deep insights into the results analysis, AP and NDCG are used as criteria to conduct the evaluation, while considering the ranking position of the services. It is clear to see that CLSM has the best scores of 1.0000 under the four metrics of AP, including AP@5, AP@10, AP@15, and AP@20, in Table 7. However, it does not present the best performance in the results of P. As k increases from 5 to 20, there is a dramatic decrease in P. More specifically, the value of P@5 reaches 0.6000, while the P@20 value is dropped to 0.1500. The main reason could lie within the differences during the calculation process between P and AP.

To facilitate the understanding of the differences, we present a specific example for the further explanations. We assume that for the specific query Q, ten services have been retrieved. Among these services, three of them are relevant and are ranked as No. 1, No. 2, and No. 3. Based on the formulas in (5) and (6), the values of P and AP can be obtained (shown below). Although three services are relevant to the given query, the AP achieves the high score of 1, which is larger than that of P (P=0.3). Similarly, such differences can also be found in the results of LDA. The value of AP@5 achieves 1.0000, but for P@5, the value is 0.4000, $P(Q)$ and $AP(Q)$ as shown at the bottom of the page.

With respect to the evaluation by using NDCG, the results of the proposed method presented in Table 7 show the best performance. The values under different settings of k are all over 0.95, which are greater than the others. Additionally, it is interesting to see that the NDCG@5 values of LDA and CLSM are 1.0000, but the corresponding P@5 values are only 0.4000 and 0.6000. This finding could be affected by the fact that the retrieved top 5 services of LDA and CLSM are ranked in line with the relevance levels. Therefore, in the process of the NDCG calculation, the value of DCG is the same as IDCG. This circumstance causes the NDCG@5 to achieve the best score of 1.0000, although the P@5 value is not good enough. However, in comparison with AP, the NDCG values show more accuracy since the grades of the relevance are set in a more fine-grained manner and it takes more factors into consideration during the calculation.

Furthermore, the results of Q2 and Q3 are presented in Tables 8 and 9, respectively. As shown in Table 8, the values of AP and NDCG for W2v-CNN are 1.0000, which are fairly different from the values of P. Having a close look at the service recommendation lists in Table 5, this study found that only the top 1 service is related to Q2, and the remaining four services have no relevance to Q2. This result indicates that achieving the best value could occur in either the best case or the worst one. Hence, we believe that it is vital to perform a comprehensive comparison and make an accurate analysis. Moreover, the values of P@5 and AP@5 for Doc2v are 0.0000 since the retrieved top 5 services have no relevance to Q2. Note that the values of the results shown in Table 8 are smaller than the results in Tables 7 and 9, which suggests that the retrieved services have little or no relevance with the query. A possible explanation could be that the dataset is not large enough to support a balanced performance. In addition, the maldistribution of the different domains of services could have an effect on the retrieval results for different queries.

Within the comparisons of all of the methods under the same evaluation metric, the results of the proposed ELMo-CNN-based method reach higher scores and present small fluctuations as the value of k increases. In the comparison of each method across multiple metrics, the results of P, AP and NDCG in the proposed ELMo-CNN-based method show consistent results, which indicates great efficiency. Therefore, these findings can imply the validation of our ELMo-CNN-based discovery method.

Figs. 5 and 6 plot the MAP and average NDCG of the service recommendation lists using different discovery methods, with k ranging from 5 to 20 at intervals of 5 for the three

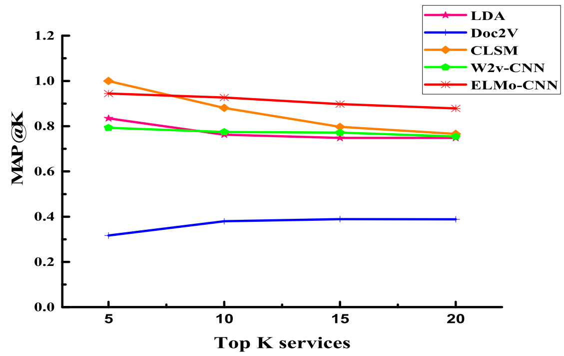

FIGURE 5. The comparison of MAP@K using different methods.

queries. As depicted in Fig. 5, our method shows the greatest accuracy on Top 10, 15, and 20. However, on the Top 5, the values of CLSM achieve the best score of 1.0000, which is higher than the score of 0.9444 in the proposed method. This finding can be explained by the fact that only the few top services listed are generated by the CLSM match with the queries, whereas the remaining services have little or no relevance. Furthermore, LDA and W2v-CNN place second with the results of 0.8347 and 0.7931, respectively. These are followed by Doc2v with the result of 0. 3167.

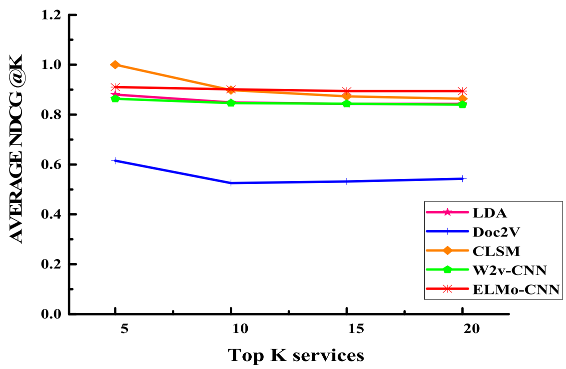

FIGURE 6. The comparison of Average NDCG@K using different methods.

Fig. 6 presents the comparison results of the average NDCG@k using different methods. Interestingly, the MAP value of Doc2v is on the rise as k increases. However, the value of the average NDCG declines. This finding could occur because the highly relevant services might not be ranked at higher positions in the recommendation list. At the same time, unlike AP using a binary value, NDCG allows the relevance to be the real value and takes a more fine-grained level for evaluation, which could decrease the impact on the results. Therefore, it is important to consider the differences of the metrics when comparing the performances of the methods under these metrics.

$$P(Q) = \frac{3}{10} = 0.3$$

$$AP(Q) = \frac{\frac{1}{1}*1+\frac{2}{2}*1+\frac{3}{3}*1+\frac{3}{4}*0+\frac{3}{5}*0+\frac{3}{6}*0+\frac{3}{7}*0+\frac{3}{8}*0+\frac{3}{9}*0+\frac{3}{10}*0}{3} = 1$$

With regard to the fluctuation range of each method, our proposed method shows great stability with the least reduction of 7% for MAP and 1.8% for average NDCG. W2v-CNN places next, with a reduction of 9.3% and 2.8% for MAP and average NDCG, respectively. Next is LDA, which reduces by 10.4% and 4.2%. CLSM is approximately 23.4% and 13.7%. In particular, Doc2v has a growth of 23% for MAP and a 14.6% reduction for average NDCG. These findings indicate that among other approaches, the expansion of the retrieval range has an effect on the discovery performance by using the proposed ELMo-CNN-based method.

TABLE 10. The comparison of MAE and RMSE for the three queries.

Methods	MAE			RMSE		
	Q1	Q2	Q3	Q1	Q2	Q3
LDA	1.8500	1.8000	1.7000	2.1095	2.0493	1.9493
Doc2v	1.6500	2.0000	1.7500	1.9621	2.2136	2.0371
CLSM	1.7500	1.7000	1.6000	2.0372	2.0493	1.9493
W2v-CNN	1.4000	2.1000	1.2500	1.7320	2.3021	1.5968
ELMo-CNN	**0.8500**	**1.4000**	**0.8000**	**1.0723**	**1.8165**	**1.0954**

To further validate the effectiveness of the proposed method, an error analysis of MAE and RMSE for the three queries is conducted, and their results are presented in Table 10. As shown in Table 10, our ELMo-CNN-based discovery method shows significant advantages among the other approaches. More specifically, the proposed method achieves the lowest amount of MAE and RMSE for each query (highlighted in bold). In particular, the results of MAE and RMSE in our method reach 0.8500 and 1.0723 for Q1, which are approximately 50% and 40% lower than the other comparative approaches. Similarly, for Q2 and Q3, our proposed method has the lowest error with scores of 1.4000 and 1.8165 and the scores of 0.8000 and 1.0954, respectively. Such results are consistent with the comparison of retrieved service descriptions, which shows that the proposed method obtains better performance.

To summarize, the results of this study show the great performance of our ELMo-CNN-based discovery method, which suggests that the integration of the ELMo representation and the Convolutional Neural Network can generate great efficiency for service discovery. Moreover, the results of the comparisons within the various metrics present the differences. Therefore, it is vital to select an appropriate criterion to conduct the evaluation.

V. CONCLUSIONS

Web service discovery is increasingly drawing the attention of practitioners and academics. Evidence from the previous studies indicates that there is a need to develop an efficient service discovery method to facilitate the improvement of discovery performance. Therefore, this study aims to make use of the relevant text matching approaches to achieve the precise Web service discovery. Furthermore, the following research questions are answered: (1) *What methods can obtain the word representations that capture the sufficient syntactic and semantic information and suit the dynamic context*? (2) *How does the method learn the matching relationships between the descriptions of the service and query at a lexical level during the service discovery process*? (3) *What refinements can be made to improve the performance for service discovery according to the specific characteristics of the neural matching models*?

More specifically, we leverage the pretrained ELMo model to obtain the word representations. In doing so, the semantic information can be captured from the descriptions of the services and queries, and the polysemous words can also be identified to be suitable for the dynamic context. This approach is beneficial for reacting to increasingly diverse service descriptions. Second, the word representations are used to constitute the input similarity matrix that can clearly characterize the matching relationship between the descriptions of the service and query at a lexical level. Additionally, the iterative performance of the convolution and pooling operations within the CNN can capture and retain the useful similarity features to receive a more accurate similarity score, which is helpful toward improving an understanding of the matching process. Finally, then considering the specific characteristics of representation-based and interaction-based neural matching models, we combine the ELMo representation and CNN to address a semantic similarity measure by integrating the multiple stages of the matching process for service discovery. It can largely promote the efficiency of the generated representations and the accuracy of the similarity calculation, which contributes to the improvement of the service discovery performance.

Furthermore, our experimental results show that our ELMo-CNN-based service discovery method presents greater performance compared with both representation-based approaches, such as Doc2v, and interaction-based approaches, such as W2v-CNN, in this study. These findings demonstrate that our service discovery method can provide better understanding of the critical steps within the process of service discovery. In addition, it points out the important challenges in providing an effective matching approach by showing the performance of the representation generation and the similarity calculation for service discovery.

Moreover, the important implications of this study are summarized. First, this study considered that the methods used in the text matching field can also be applied for service discovery because the functionalities of the Web services and users' requirements are commonly described in natural language. This approach could imply that except for the text matching field, we could broaden the research horizon and utilize the research approaches in relevant fields, such as intelligent question-answer, to promote the performance of the current work. Second, the ELMo model is invoked by adopting the key idea of transfer learning. It is worthwhile to further develop the simple and reliable methods since transfer learning is helpful to maintaining the stability of the results and reducing the operational complexity toward a small amount of the training data. Third, the matching process is conducted based on the similarity matrix of pairs of words, which addresses the multiple stages of the matching process and provides a better solution to capture the sufficient matching information for calculating the final similarity.

This study has some limitations. First, this work selected only one dataset with three test queries to experimentally validate the proposed method, which explains the exploratory nature of our work. To perform a more comprehensive assessment to draw representative conclusions, further research could be conducted on a wider range of datasets with more test queries. Another limitation is that the description of the Web services and queries are manually extracted from the corresponding owl documents. When addressing the large scale resources, the manual approach will be time consuming and will influence the accuracy of the data processing. It would be interesting to conduct a study that implements the validation on an automatic resources processing method. The results could provide deeper insights into the effectiveness of web service descriptions, extraction, and discovery. This study is the first step, and additional studies will be conducted. For example, we will focus on various aspects of improving the service discovery performance, such as integrating the matching of interface signatures and QoS attributes, optimizing the CNN model, or developing prototypes for practical validation. The findings would be valuable for developing more intelligent and effective methods for service discovery.

REFERENCES

[1] J. Wang, P. Gao, Y. Ma, K. He, and P. C. K. Hung, "A Web service discovery approach based on common topic groups extraction," *IEEE Access*, vol. 5, pp. 10193–10208, 2017.

[2] F. Chen, C. Lu, H. Wu, and M. Li, "A semantic similarity measure integrating multiple conceptual relationships for Web service discovery," *Expert Syst. Appl.*, vol. 67, pp. 19–31, Jan. 2017.

[3] L. Duan and H. Tian, "Collaborative Web service discovery and recommendation based on social link," *Future Internet*, vol. 9, no. 4, p. 63, Oct. 2017.

[4] S. Rangarajan, "Qos-based Web service discovery and selection using machine learning," *ICST Trans. Scalable Inf. Syst.*, vol. 5, no. 17, May 2018, Art. no. 154809.

[5] N. Zhang, J. Wang, Y. Ma, K. He, Z. Li, and X. Liu, "Web service discovery based on goal-oriented query expansion," *J. Syst. Softw.*, vol. 142, pp. 73–91, Aug. 2018.

[6] J. J. Jung, "Service chain-based business alliance formation in service-oriented architecture," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2206–2211, Mar. 2011.

[7] M. Norman and W. Ingo, "Information gathering for semantic service discovery and composition in business process modeling," in *Proc. LNBIP*, vol. 10, 2008, pp. 46–60.

[8] F. Chen, M. Li, H. Wu, and L. Xie, "Web service discovery among large service pools utilising semantic similarity and clustering," *Enterprise Inf. Syst.*, vol. 11, no. 3, pp. 452–469, Mar. 2017.

[9] M. Fang, D. Wang, Z. Mi, and M. S. Obaidat, "Web service discovery utilizing logical reasoning and semantic similarity," *Int. J. Commun. Syst.*, vol. 31, no. 10, p. e3561, Jul. 2018.

[10] N. Zhang, Z. Li, K. He, and J. Wang, "WSGM-SD: An approach to RESTful service discovery based on weighted service goal model," *Chin. J. Electron.*, vol. 25, no. 2, pp. 256–263, Mar. 2016.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," Cornell Univ., Ithaca, NY, USA, Tech. Rep., Sep. 2013.

[12] B. Hu, Z. D. Lu, H. Li, and Q. C. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2042–2050.

[13] M. Bryan, B. James, C. Xiong, and S. Richard, "Learned in translation: Con-textualized word vectors," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6294–6305.

[14] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., (Long Papers)*, vol. 1, 2018, pp. 1–11.

[15] L. X. Meng and Y. Li, "M2S-Net: Multi-modal similarity metric learning based deep convolutional network for answer selection," Cornell Univ., Ithaca, NY, USA, Tech. Rep., Mar. 2018.

[16] Y. R. Qin, Q. Z. Sheng, J. G. F. Nickolas, D. Schahram, H. Wang, and V. V. Athanasios, "When things matter: A data-centric view of the Internet of Things," Cornell Univ., Ithaca, NY, USA, Tech. Rep., Jul. 2014.

[17] N. Wang, J. Fu, B. K. Bhargava, and J. Zeng, "Efficient retrieval over documents encrypted by attributes in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2653–2667, Oct. 2018.

[18] Z. Shao, W. Zhou, X. Deng, M. Zhang, and Q. Cheng, "Multilabel remote sensing image retrieval based on fully convolutional network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 318–328, Jan. 2020.

[19] A. A. Liu, W. Z. Nie, Y. Gao, and Y. T. Su, "Multi-modal clique-graph matching for view-based 3D model retrieval," *IEEE T. Image Process.*, vol. 25, no. 5, pp. 2013–2116, Mar. 2016.

[20] H. Wang and A. Zhou, "Service discovery based on user latent intentions," *Chin. J. Electron.*, vol. 25, no. 5, pp. 841–847, Sep. 2016.

[21] Z. Cong, A. Fernandez, H. Billhardt, and M. Lujak, "Service discovery acceleration with hierarchical clustering," *Inf. Syst. Frontiers*, vol. 17, no. 4, pp. 799–808, Aug. 2015.

[22] J. Sangers, F. Frasincar, F. Hogenboom, and V. Chepegin, "Semantic Web service discovery using natural language processing techniques," *Expert Syst. Appl.*, vol. 40, no. 11, pp. 4660–4671, Sep. 2013.

[23] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web services description language (WSDL) 1.1," World Wide Web Consortium, Cambridge, MA, USA, Tech. Rep. wsdl-20010315, 2001.

[24] A. De Renzis, M. Garriga, A. Flores, A. Cechich, C. Mateos, and A. Zunino, "A domain independent readability metric for Web service descriptions," *Comput. Standards Interface*, vol. 50, pp. 124–141, Feb. 2017.

[25] J. Cardoso, *Semantic Web Services: Theory, Tools and Applications*. Hershey, PA, USA: IGI Global, 2007, pp. 191–216.

[26] H. H. Wang, N. Gibbins, T. R. Payne, and D. Redavid, "A formal model of the semantic Web service ontology (WSMO)," *Inf. Syst.*, vol. 37, no. 1, pp. 33–60, Mar. 2012.

[27] A. Fellah, M. Malki, and A. Elci, "A similarity measure across ontologies for Web services discovery," *Int. J. Inf. Technol. Web Eng.*, vol. 11, no. 1, pp. 22–43, Jan. 2016.

[28] S. Bechhofer, F. V. Harmelen, J. Hendler, I. Horrocks, D. L. Mcguinness, P. F. Patel-Schneider, and L. A. Stein, "OWL Web ontology language reference," *W3C Recommendation*, vol. 10, no. 2, pp. 1–53, 2004.

[29] J. Domingue, D. Roman, and M. Stollberg, "Web service modeling ontology (WSMO)—An ontology for semantic Web services," in *Proc. W3C Workshop Frameworks Semantics Web Services*, Innsbruck, Austria, 2005, pp. 9–10.

[30] T. Louge, M. H. Karray, B. Archimède, Z. Maamar, and M. Mrissa, "Semantic Web services composition in the astrophysics domain: Issues and solutions," *Future Gener. Comput. Syst.*, vol. 90, pp. 185–197, Jan. 2019.

[31] Y. Hao, Y. Zhang, and J. Cao, "Web services discovery and rank: An information retrieval approach," *Future Gener. Comput. Syst.*, vol. 26, no. 8, pp. 1053–1062, Oct. 2010.

[32] C. Wu, "WSDL term tokenization methods for IR-style Web services discovery," *Sci. Comput. Program.*, vol. 77, no. 3, pp. 355–374, Mar. 2012.

[33] W. Lu, Y. Cai, X. Che, and Y. Lu, "Joint semantic similarity assessment with raw corpus and structured ontology for semantic-oriented service discovery," *Pers. Ubiquitous Comput.*, vol. 20, no. 3, pp. 311–323, May 2016.

[34] N. Zhang, J. Wang, and Y. Ma, "Mining domain knowledge on service goals from textual service descriptions," *IEEE Trans. Services Comput.*, vol. 13, no. 3, pp. 488–502, May 2020.

[35] J. M. García, D. Ruiz, and A. Ruiz-Cortés, "Improving semantic Web services discovery using SPARQL-based repository filtering," *J. Web Semantics*, vol. 17, pp. 12–24, Dec. 2012.

[36] P. Rodriguez-Mier, C. Pedrinaci, M. Lama, and M. Mucientes, "An integrated semantic Web service discovery and composition framework," *IEEE Trans. Services Comput.*, vol. 9, no. 4, pp. 537–550, Jul. 2016.
[37] H. Naim, M. Aznag, M. Quafafou, and N. Durand, "Probabilistic approach for diversifying Web services discovery and composition," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2016, pp. 73–80.
[38] B. Hu, Z. Zhou, and Z. Cheng, "Web services recommendation leveraging semantic similarity computing," *Procedia Comput. Sci.*, vol. 129, pp. 35–44, Jan. 2018.
[39] S.-P. Ma, C.-W. Lan, and C.-H. Li, "Contextual service discovery using term expansion and binding coverage analysis," *Future Gener. Comput. Syst.*, vol. 48, pp. 73–81, Jul. 2015.
[40] A. Heppner, A. Pawar, D. Kivi, and V. Mago, "Automating articulation: Applying natural language processing to post-secondary credit transfer," *IEEE Access*, vol. 7, pp. 48295–48306, 2019.
[41] Y. Song, Q. V. Hu, and L. He, "P-CNN: Enhancing text matching with positional convolutional neural network," *Knowl.-Based Syst.*, vol. 169, pp. 67–79, Apr. 2019.
[42] C. D. Manning, P. Raghavan, and H. Schütze, "Scoring, term weighting and the vector space model," in *Introduction to Information Retrieval*. Cambridge, U.K.: CUP, 2008, ch. 6, sec. 3, pp. 120–126.
[43] L. Boytsov, D. Novak, Y. Malkov, and E. Nyberg, "Off the beaten path: Let's replace term-based retrieval with k-NN search," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 1099–1108.
[44] H. Yao, H. Liu, and P. Zhang, "A novel sentence similarity model with word embedding based on convolutional neural network," *Concurrency Comput., Pract. Exper.*, vol. 30, no. 23, p. e4415, Jan. 2018.
[45] S. Poria, I. Chaturvedi, E. Cambria, and F. Bisio, "Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 4465–4473.
[46] Z. Jingrui, W. Qinglin, L. Yu, and L. Yuan, "A method of optimizing LDA result purity based on semantic similarity," in *Proc. 32nd Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, May 2017, pp. 361–365.
[47] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.
[48] H. T. Nguyen, P. H. Duong, and E. Cambria, "Learning short-text semantic similarity with word embeddings and external knowledge sources," *Knowl.-Based Syst.*, vol. 182, Oct. 2019, Art. no. 104842.
[49] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[50] Z. Gao, H. Xue, and S. Wan, "Multiple discrimination and pairwise CNN for view-based 3D object retrieval," *Neural Netw.*, vol. 125, pp. 290–302, May 2020.
[51] Z. Gao, H.-Z. Xuan, H. Zhang, S. Wan, and K.-K.-R. Choo, "Adaptive fusion and category-level dictionary learning model for multiview human action recognition," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9280–9293, Dec. 2019.
[52] W. Lu, Y. Cai, X. Che, and K. Shi, "Semantic similarity assessment using differential evolution algorithm in continuous vector space," *J. Vis. Lang. Comput.*, vol. 31, pp. 246–251, Dec. 2015.
[53] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, "A study on similarity and relatedness using distributional and WordNet-based approaches," in *Proc. Human Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics NAACL*, 2009, pp. 19–27.
[54] A. Alves, A. Ferrugento, M. Lourenço, and F. Rodrigues, "ASAP: Automatic semantic alignment for phrases," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 104–108.
[55] D. Kiela and S. Clark, "A systematic study of semantic vector space model parameters," in *Proc. 2nd Workshop Continuous Vector Space Models Their Compositionality (CVSC)*, 2014, pp. 21–30.
[56] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. F. Gao, "Deep learning based text classification: A comprehensive review," Cornell Univ., Ithaca, NY, USA, Tech. Rep., Apr. 2020.
[57] W. Zhao, H. Peng, S. Eger, E. Cambria, and M. Yang, "Towards scalable and reliable capsule networks for challenging NLP applications," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1549–1559.
[58] C. V. Gysel, M. de Rijke, and E. Kanoulas, "Neural vector spaces for unsupervised information retrieval," *ACM Trans. Inf. Syst.*, vol. 36, no. 4, pp. 1–25, Oct. 2018.
[59] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for ad-hoc retrieval," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 55–64.
[60] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for Web search using clickthrough data," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. CIKM*, 2013, pp. 2333–2338.
[61] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "Learning semantic representations using convolutional neural networks for Web search," in *Proc. 23rd Int. Conf. World Wide Web WWW Companion*, 2014, pp. 373–374.
[62] Z. Lu and H. Li, "A deep architecture for matching short texts," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1367–1375.
[63] E. Al-Masri and Q. H. Mahmoud, "Discovering the best Web service: A neural network-based solution," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2009, pp. 4250–4255.
[64] Y. Zhang, G. Xu, Y. Wang, X. Liang, L. Wang, and T. Huang, "Empower event detection with bi-directional neural language model," *Knowl.-Based Syst.*, vol. 167, pp. 87–97, Mar. 2019.
[65] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[66] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2 Vec," *Inf. Sci.*, vol. 477, pp. 15–29, Mar. 2019.
[67] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage. CIKM*, 2014, pp. 101–110.
[68] K. Hu, H. Wu, K. Qi, J. Yu, S. Yang, T. Yu, J. Zheng, and B. Liu, "A domain keyword analysis approach extending term frequency-keyword active index with Google Word2 Vec model," *Scientometrics*, vol. 114, no. 3, pp. 1031–1068, Mar. 2018.

ZHAO HUANG received the Ph.D. degree in information systems and computing from Brunel University, U.K. He was a Postdoctoral Research Fellow WITH the Telfer School of Management, University of Ottawa, Ottawa, ON, Canada. He is currently an Associate Professor with Shaanxi Normal University. His main research interests include information systems, human–computer interaction, and intelligent recommendation systems.

WEI ZHAO is currently pursuing the master's degree with the Shaanxi Normal University. Her research interests include text mining, service discovery, and intelligent recommendation systems.