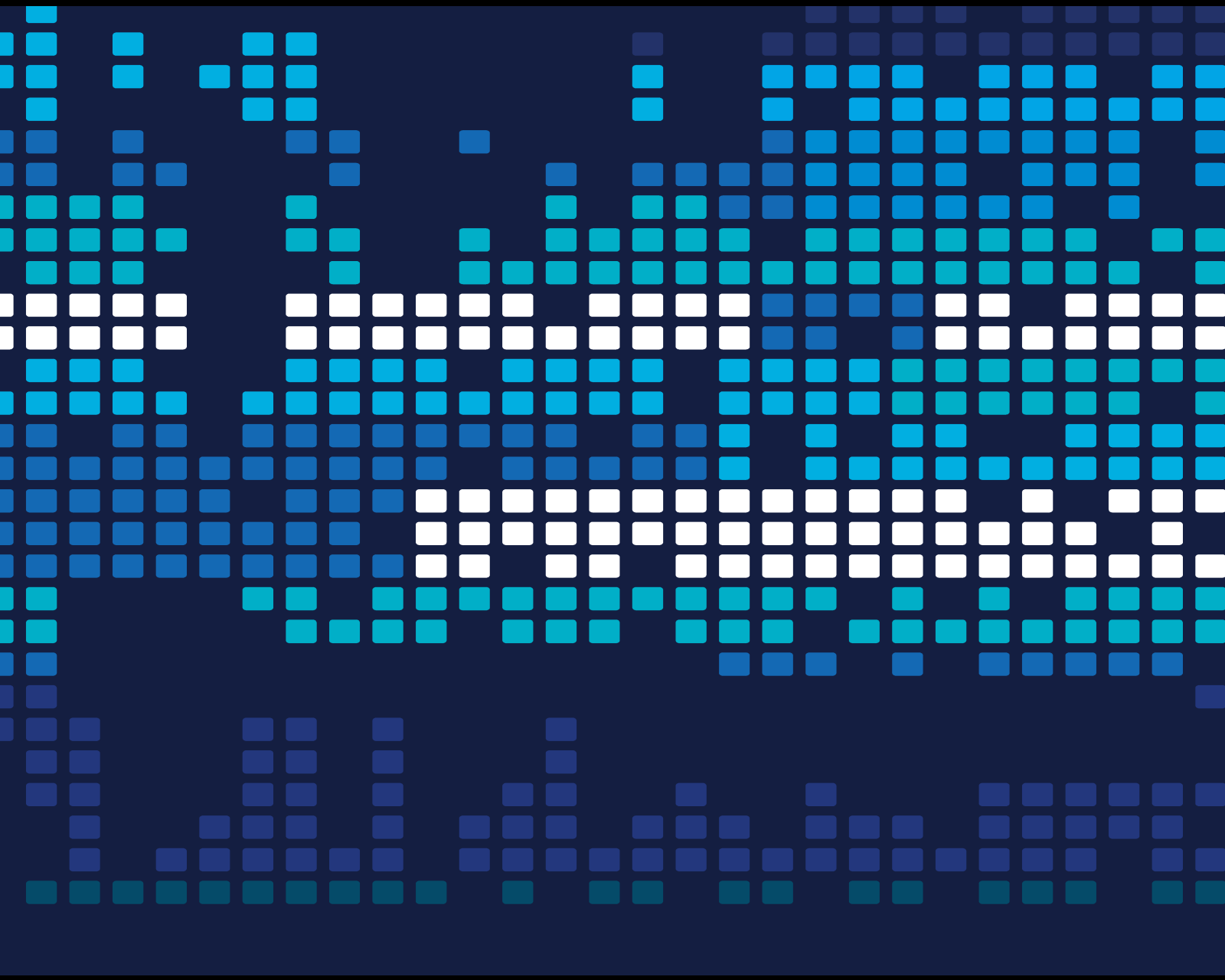
Scientific Programming

Novel Tools for the Management, Representation, and Exploitation of Textual Information

Lead Guest Editor: David Ruano-Ordás
Guest Editors: José R. Méndez, Vitor Fernandes, and Guillermo Suárez-Tangil



Novel Tools for the Management, Representation, and Exploitation of Textual Information

Scientific Programming

Novel Tools for the Management, Representation, and Exploitation of Textual Information

Lead Guest Editor: David Ruano-Ordás
Guest Editors: José R. Méndez, Vitor Fernandes, and Guillermo Suárez-Tangil

Contents

Hindawi
Scientific Programming
Volume 2021, Article ID 9781923, 3 pages
https://doi.org/10.1155/2021/9781923

Editorial

Novel Tools for the Management, Representation, and Exploitation of Textual Information

David Ruano-Ordás,[1,2,3] **Jose R. Méndez**,[1,2,3] **Vítor Basto Fernandes**,[4] **and Guillermo Suárez-Tangil**[5,6]

[1]*Department of Computer Science, University of Vigo, ESEI-Escuela Superior de Ingeniería Informática, Edificio Politécnico, Campus Universitario As Lagoas s/n, 32004 Ourense, Spain*
[2]*CINBIO-Biomedical Research Centre, University of Vigo, Campus Universitario Lagoas-Marcosende, 36310 Vigo, Spain*
[3]*SING Research Group, Galicia Sur Health Research Institute (IIS Galicia Sur), SERGAS-UVIGO, 36312 Vigo, Pontevedra, Spain*
[4]*Instituto Universitário de Lisboa (ISCTE-IUL), University Institute of Lisbon, ISTAR-IUL, Av. Das Forças Armadas, 1649-026 Lisboa, Portugal*
[5]*IMDEA Networks Institute, Av. Del Mar Mediterraneo, 22, Leganes, Spain*
[6]*Department of Informatics, King's College London, Faculty of Natural and Mathematical Science, Strand Campus, London, UK*

Correspondence should be addressed to David Ruano-Ordás; drordas@uvigo.es

Received 28 July 2021; Accepted 28 July 2021; Published 15 August 2021

Over the last decade, the explosive growth of social media and instant message applications together with the improvement of computer performance, networking infrastructures, and storage capabilities have led to the advent of the information age. Particularly, most people in industrialized countries have permanent and unlimited access to the Internet via mobile devices (smartphones, tablets, etc.). This infrastructure allows users to generate and send massive data to Internet servers from everywhere [1].

As long as human communications are made through language, most of the information gathered from mobile devices is textual. Common examples are instant messages, e-mails sent/received, updates sent to microblogs and/or social networks, opinions on products/apps, etc. This scenario has propitiated the massive dissemination and collection of massive and diverse textual information from multiple (and almost unlimited) data sources. However, the deluge of raw text data is neither meaningful nor useful without the use of proper methodologies to detect and extract valuable knowledge. To this end, we have selected some works that clearly contribute by providing relevant ideas, methodologies, and models on the topic of this special issue. A summary of these works and their specific findings are included below.

Knowing relevant information about Internet users in the current digital era is a crucial competitive advantage for industries. Consequently, several works have been developed and introduced in recent years focusing on the management, representation, and exploitation of the knowledge available from textual sources. One of the most interesting works in this line is the profiling and clustering of users using information shared through social networks [2]. Concretely, they propose a data collection framework to obtain specific data on individuals to explore user profiles and identify segments based on these profiles. This information is particularly relevant in the customization of user-oriented websites (advertisements, news referrals, etc.). Next, the work presented in [3] addresses the age prediction problem by combining social media-specific metadata and language-related features. To accomplish this task, the authors combine (i) part-of-speech *N*-gram features, (ii) stylometry features (average sentence length, average word length, etc.), and (iii) features from lexicons that correlate words/phrases with specific age and sentiment scores together using deep learning schemes via Keras (available at https://keras.io) framework achieving a good performance.

Another interesting area which is also related with social media textual data exploitation is the detection of breaking

news and trending histories in social networks to avoid spreading rumours (unverified stories or statements) or fake news (misleading information presented as news). These kinds of situations produce serious damage in different areas such as personal or professional life, corporate image of a company, or even a stock market turmoil. The work described in [4] extracts word-embedding features from social networks to train a deep learning model (recurrent neural network) to automatically identify rumours and mitigate topic shift issues. Concerning the fake news detection, we can highlight the works described in [5, 6]. The former proposes a fake news detector (FNDNET) based on the usage of a deep convolutional neural network (CNN). Achieved results applied over the Kaggle fake news dataset show that FNDNet clearly outperforms the results gathered by other well-known alternatives. The latter explores different textual properties (useful to distinguish between fake and real contents) to train a combination of different machine learning algorithms using various ensemble methods. Experimental evaluation carried out over four different real-world datasets confirms the superior performance of the proposed ensemble in comparison to individual learners.

From a medical perspective, textual information posted in social networks could be an important thermometer to both detect and suggest medical diseases/treatments and measure the quality of healthcare services. Particularly, in [7], natural language processing (NLP) and sentiment analysis are used to analyse patient experiences shared through the Internet to assess healthcare performance. Moreover, the work described in [8] presents how latent Dirichlet allocation (LDA) and random forests (RF) were adequately combined to find latent topics of healthcare and show the utility of social media forums to automatically detect healthcare issues in patients.

Another important area lies in the automatic identification, detection of interpersonal and gender-based violence. In this sense, the work of [9] proposed a methodology to detect and associate fake profiles on Twitter used for defamatory purposes based on analysing the content of the comments generated by troll and victim profiles. In their methodology, they used text, time of publication, language, and geolocation as features. They compared different machine learning (ML) classifiers including random forests, J48, K-nearest neighbour (KNN), and sequential minimal optimization (SMO) for assessing the probability of a user being the author of a tweet. The experimentation carried out used the false-positive/false-negative ratio and area under receiver operating characteristic (AUC) curves to demonstrate the suitability of the proposal.

Finally, the identification of influencers in social networks has also been addressed [10]. Identifying hot blogs and opinion leaders allows marketers to determine if the opinions shared are favourable to sell their products. The work of Li and Du introduces a framework to identify opinion leaders using the information retrieved from blog content, authors, readers, and their relationships. Blog contents are used to automatically learn an ontology. This ontology is used to measure expertise, find readers, and assess relationships between readers and blog authors. This data is used to find hot blogs and assess the influence of bloggers.

This special issue brings together several papers showing different utilities of text mining and NLPA. It comprises four high-quality works submitted by researchers from China and India, which were selected from the submitted ones. In general, published studies address the following problems: (i) clustering web services, (ii) identification of hotspots (current hot topics), (iii) prediction of user behaviour for the early fetch of interesting web pages, and (iv) the application of named entity recognition (NER) in medical documents written in Chinese.

The first study included in the special issue authored by C. Shan and Y. Du [11] presents a method to automatically group similar web services. Web services are usually Representational State Transfer (REST) Application Programmers Interfaces (API) and provide a collection of tags that are used for clustering. To this end, the authors propose two algorithms. The former is in charge of computing the semantic similarity between the tags of different online available APIs by using the WordNet lexicon database. The latter one is responsible for grouping the APIs according to the values previously computed. To analyse the performance of the proposal, the authors grouped the services available in ProgrammableWeb (available at https://www.programmableweb.com) and measured the performance using recall, precision, and F-score. The proposal outperforms the other five popular and classical clustering approaches.

The work presented by H.R. Cao et al. [12] proposes a hybrid solution for identifying online hotspots, assessing their importance, and enabling their monitoring. They integrate different mechanisms for filtering invalid user posts and replies and design an algorithm to extract keywords from hotspots. Experimental evaluation showed that the method could effectively filter out invalid data, improve the representation of datasets, and reflect changes in hotspot trends.

The proposal included by S. Setia et al. [13] introduced a methodology to accurately model the behaviour of web users. To this end, web browsers can fetch web pages in the background before the user explicitly demands them to improve the experience. The model used to predict the behaviour uses *N-gram* parsing and the click-counter of queries to improve the prediction of web pages. Experimental results have shown that the proposed strategy can significantly reduce the fetching time.

Finally, the work by Y. Wang et al. [14] present a new method for data augmentation based on Masked Language Model (MLM). They compare the performance of NER in the Chinese medical literature using the data augmentation method, pretraining models (such as Bidirectional Encoder Representations from Transformers (BERT), ERNIE (available at https://github.com/PaddlePaddle/ERNIE), or RoBERTa [15]), common deep learning models (such as Bidirectional Long Short-Term Memory (BiLSTM) [16]), and downstream models with different structures (FC, CRF, LSTM-CRF, and BiLSTM-CRF). Their experiments showed the utility of their data augmentation method to improve the

performance of entity recognition, which can also be used to increase the performance of pretraining models.

Despite the abovementioned works in the context of the management, representation, and exploitation of textual information, this field of computer science includes major challenges that have yet to be resolved. We sincerely hope that readers enjoy the special issue and find it worthy for understanding the real value of textual information compiled worldwide.

Conflicts of Interest

The Guest Editors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

David Ruano-Ordás acknowledges Xunta de Galicia for its support under its fellowship program (ED481D-2021/024). José R. Méndez acknowledges the funding support of the Spanish Ministry of Economy, Industry and Competitiveness (SMEIC), State Research Agency (SRA), and the European Regional Development Fund (ERDF) (Semantic Knowledge Integration for Content-Based Spam Filtering, TIN2017-84658-C2-1-R). Vitor Basto-Fernandes acknowledges FCT (Fundação para a Ciência e a Tecnologia), I.P., for its support in the context of projects UIDB/04466/2020 and UIDP/04466/2020. We would also like to thank all authors for their contributions to this special issue and the reviewers for their generous time in providing detailed comments and suggestions that helped us to improve the quality of this special issue.

David Ruano-Ordás
José R. Méndez
Vitor Basto-Fernandes
Guillermo Suárez-Tangil

References

[1] M. Anshari and Y. Alas, "Smartphones habits, necessities, and big data challenges," *The Journal of High Technology Management Research*, vol. 26, no. 2, pp. 177–185, 2015.

[2] J.-W. Van Dam and M. Van de Velden, "Online profiling and clustering of Facebook users," *Decision Support Systems*, vol. 70, no. 2, pp. 60–72, 2015.

[3] A. Pandya, M. Oussalah, P. Monachesi, and P. Kostakos, "On the use of distributed semantics of tweet metadata for user age prediction," *Future Generation Computer Systems*, vol. 102, no. 1, pp. 437–452, 2020.

[4] S. A. Alkhodair, S. H. H. Ding, B. C. M. Fung, and J. Liu, "Detecting breaking news rumors of emerging topics in social media," *Information Processing & Management*, vol. 57, no. 2, Article ID 102018, 2020.

[5] R. K. Kaliyar, A. Goswami, P. Narang, S. Sinha, and S. Sinha, "FNDNet-a deep convolutional neural network for fake news detection," *Cognitive Systems Research*, vol. 61, no. 6, pp. 32–44, 2020.

[6] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, vol. 2020, no. 10, 11 pages, Article ID 8885861, 2020.

[7] F. Greaves, D. Ramirez-Cano, C. Millett, A. Darzi, and L. Donaldson, "Harnessing the cloud of patient experience: using social media to detect poor quality healthcare: table 1," *BMJ Quality and Safety*, vol. 22, no. 3, pp. 251–255, 2013.

[8] H. Jelodar, Y. Wang, M. Rabbani, G. Xiao, and R. Zhao, "A collaborative framework based for semantic patients-behavior analysis and highlight topics discovery of alcoholic beverages in online healthcare forums," Journal of Medical Systems, vol. 44, no. 5, p. 101, 2020.

[9] P. Galán-García, J. G. de la Puerta, C. L. Gómez, I. Santos, and P. G. Bringas, "Supervised machine learning for the detection of troll profiles in twitter social network: application to a real case of cyberbullying," in *Proceedings of Advances in Intelligent Systems and Computing,International Joint Conference SOCO'13-CISIS'13-ICEUTE'13*, pp. 419–428, Salamanca, Spain, September 2014.

[10] F. Li and T. C. Du, "Who is talking? An ontology-based opinion leader identification framework for word-of-mouth marketing in online social blogs," *Decision Support Systems*, vol. 51, no. 1, pp. 190–197, 2011.

[11] C. Shan and Y. Du, "A web service clustering method based on semantic similarity and multidimensional scaling analysis," *Scientific Programming*, vol. 2021, no. 5, 12 pages, Article ID 6661035, 2021.

[12] H. Cao, X. Li, S. Lian, and C. Zhan, "A hotspot information extraction hybrid solution of online posts' textual data," *Scientific Programming*, vol. 2021, no. 4, 11 pages, Article ID 6619712, 2021.

[13] S. Setia, V. Jyoti, and N. Duhan, "HPM: a hybrid model for user's behavior prediction based on N-gram parsing and access logs," *Scientific Programming*, vol. 2020, no. 11, 18 pages, Article ID 8897244, 2020.

[14] Y. Wang, Y. Sun, Z. Ma, L. Gao, and Y. Xu, "Named entity recognition in Chinese medical literature using pretraining models," *Scientific Programming*, vol. 2020, no. 9, 9 pages, Article ID 8812754, 2020.

[15] Y. Liu, M. Ott, N. Goyal et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, https://arxiv.org/abs/1907.11692>.

[16] X. Chen, C. Ouyang, Y. Liu, and Y. Bu, "Improving the named entity recognition of Chinese electronic medical records by combining domain dictionary and rules," International Journal of Environmental Research and Public Health, vol. 17, no. 8, p. 2687, 2020.

Hindawi
Scientific Programming
Volume 2021, Article ID 6661035, 12 pages
https://doi.org/10.1155/2021/6661035

Research Article

A Web Service Clustering Method Based on Semantic Similarity and Multidimensional Scaling Analysis

Chuang Shan and Yugen Du

Shanghai Key Laboratory of Trustworthy Computing, School of Software Engineering, East China Normal University, Shanghai 200062, China

Correspondence should be addressed to Yugen Du; ygdu@sei.ecnu.edu.cn

Received 5 November 2020; Revised 7 February 2021; Accepted 22 April 2021; Published 5 May 2021

Academic Editor: David Ruano-Ordás

Clustering web services is an effective method to solving service computing problems. The key insight behind it is to extract the vectors based on the service description documents. However, the brevity of natural language service description documents typically complicates the vector construction process. To circumvent the difficulty, we propose a novel web service clustering method to vectorize documents based on the semantic similarity, which can be calculated via WordNet and multidimensional scaling (WMS) analysis. We utilize the dataset from the ProgrammableWeb to conduct extensive experiments and achieve prominent advances in precision, recall, and F-measure.

1. Introduction

Through the rapid development of Internet technology [1], clustering web services has become an effective method to solving service discovery [2–4], service composition [5, 6], and service recommendation [7]. Firstly, there are increasing enterprises and institutions that encapsulate software functions or data into web services and publish them to the network. For instance, the number of services in ProgrammableWeb has grown from fewer than 3,000 in 2011 to more than 20,000 by 2020, which remarkably increases the difficulty of managing web services. Secondly, when users query the web services they need, the service discovery system generally searches all related web services and returns ordered ones, where the ranking index is mainly based on the relevance to the query. However, it is intractable to search through the entire whole web services space and obtain accurate results. According to the work made by Zhang et al., clustering web services can improve the performance of service discovery by reducing the search space [2]. Thirdly, service composition is proposed to select appropriate web services from the repository to build functional web services. However, the scale of the repository will influence the efficiency of finding and sorting multiple web services with various functions. Clustering web services that matches the clusters to the requirements of developers can successfully alleviate this dilemma in the light of the research results of Xia et al. [5].

The premise of clustering web services is to extract vectors corresponding to service description documents, which are mainly constructed based on keyword or semantic features. Unfortunately, despite the effectiveness of clustering web services towards a variety of web service tasks, its applicability is hindered by extracting vectors from natural language service description documents. The service description document is an important basis for clustering web service, which is commonly implemented by Web Services Description Language (WSDL) document or natural language web service description document. Though WSDL document written in Extensible Markup Language (XML) can offer plentiful convenient functions such as describing the service in combination with Web Ontology Language (OWL), its construction procedure is quite complex. Therefore, some companies and institutions, such as ProgrammableWeb, leverage natural language to describe web services to generate succinct service description documents,

where each keyword appears almost once. However, the brevity of these documents leads to extra problems when clustering web services extracts' two types of target features. Specifically, extraction of keyword features highly depends on the frequency of keyword occurrence. Similarly, the corpus of service description documents can hardly establish so that the corresponding probabilistic topic model is difficult to construct to extract the semantic features that refers to the probability distribution of a document on different topics.

This paper proposes a novel approach that constructs vectors via differences between documents instead of document features. We mainly cluster the web services described in natural language. The operation object is the natural language service description document, and the dataset is from ProgrammableWeb. The main contributions of this paper are based on (1) designing of an algorithm to calculate the similarity between documents, (2) proposing a methodology to convert similarity data into distance data, which is an important prerequisite for multidimensional scaling analysis, and (3) the implementing principal component analysis (PCA) methods on the vectors corresponding to the service documents to determine the appropriate clustering algorithm.

The rest of our work is structured as follows. Section 2 compares existing work. Section 3 introduces the study materials. We explain the study methods in detail in Section 4. We explain the experimental process in detail in Section 5. Finally, Section 6 summarizes the main conclusions and highlights future work.

2. Related Work

In this section, we will separately introduce the related works on WSDL service description document clustering and natural language service description document clustering.

2.1. WSDL Service Description Document. In the early days, there were many web services described using WSDL documents, so many scholars paid attention to the clustering of such web services. Paik and Kumara et al. used the ontology model in service clustering [8, 9], which greatly improved the service clustering effect. Some scholars used WordNet to calculate the semantic similarity [8, 10], but the algorithm they proposed is not suitable for natural language documents. We proposed an algorithm for calculating the semantic similarity between natural language documents using WordNet. In addition, some scholars also used context-aware methods to improve service clustering [9, 11]. Liang et al. used tag information in WSDL document clustering to improve the clustering effect [12]. Considering the sparse semantics of WSDL documents, Gu et al. used open data to increase semantic information before clustering [13]. Because of the too much useless information of WSDL documents, Agarwal et al. used a probability model to filter useless information before clustering [14]. Sun et al. added neural networks to service clustering [4]. In general, these methods have a common limitation, and they are not suitable for processing service documents described in natural language. For example, in literature [8], separate ontology is constructed for different "element" data, and the "element" includes <definitions>, <types>, <messages>, and <portType>. However, in natural language documents, there is no "element," so it is very difficult to construct ontology.

2.2. Natural Language Service Description Document. Because WSDL documents are too complex to construct, some companies and organizations now use natural language to describe web services. Some scholars focus on the clustering of natural language service description documents. Muth and Inkpen used the term frequency-inverse document frequency (TF-IDF) to extract keywords and then clustered web services according to keywords [15]. However, the service description documents are too short to extract keywords. Some scholars used the latent Dirichlet allocation (LDA) to build a probabilistic topic model and calculated the probability distribution of each service description document on each topic so as to achieve document vectorization and then clustered the vectors [2, 16]. The premise of LDA is to construct the unigram model. However, the corpus of service description document is too few, and the constructed unigram model is weak. Some scholars used the Word2Vec training external corpus to expand service documents to improve the effect of LDA training [17, 18]. However, the size and type of the corpus seriously affect the degree of improvement. Lizarralde used deep variational autoencoders in this work to solve this problem [3]. Cao et al. used the Doc2Vec model to train the service document dataset, converted each document into a vector, and then clustered the vectors [19]. However, there is no reference basis for the selection of vector dimensions, which increases the uncertainty of the results. Zou et al. first trained the WE-LDA model to obtain the probability-topic distribution of each document, then trained the recurrent convolutional neural network (RCNN) to obtain a fitting model from each service document to the probability-topic distribution, and finally clustered the document-feature vectors [20, 21]. However, the structure of RCNN is very complicated, the training effect of RCNN depends on adjusting the parameters, and the training results of the LDA model greatly increase the uncertainty of RCNN. So, it is very difficult to get a suitable model by adjusting parameters. In short, the problem of these methods comes from the uncertainty caused by mining service document features. We noticed that it is difficult to extract features from short documents, but it is easier to compare the differences between short documents, so we use WordNet to quantify document differences and then use multidimensional scaling analysis to construct vectors corresponding to the documents and finally cluster the vectors. In our method, only very few parameters need to be adjusted and our work on adjusting parameters has a theoretical and experimental basis.

3. Study Materials

The experimental data in this paper come from the ProgrammableWeb website. This article uses the WordNet database to calculate semantic similarity, and we will introduce them in detail below.

3.1. ProgrammableWeb. ProgrammableWeb is an information and news source about the Web as a programmable platform. It is a subsidiary of MuleSoft and has offices in San Francisco, CA. The website publishes a repository of web APIs, mashups, and applications and has documented over 22000 open web APIs and thousands of applications in October 2020. It has been called the "journal of the API economy" by TechCrunch [22]. The data in ProgrammableWeb mainly include category, description document, tag, and calling method (see website https://www.programmableweb.com/) (see Figure 1). This paper uses description documents as the main body for service clustering. "Tag" is the auxiliary information given by web service developers, which helps us to preprocess service documents. "Category" is the classification given by web service developers, and we use it as the evaluation index of clustering.

3.2. WordNet. WordNet is an English dictionary established and maintained by the Cognitive Science Laboratory of Princeton University [23]. Because it contains semantic information, it is different from a dictionary in the usual sense. WordNet groups the entries according to their meanings. Each group of entries with the same meaning is called a *Synset*. WordNet provides a short, summary definition for each *Synset* and records the semantic relationship between different *Synsets*. A word may have multiple meanings, which are in different *Synsets* (see Table 1).

Synset contains a variety of semantic relations, such as upper and lower relation, antisense relation, and whole and part relation (see Figure 2). Based on these relationships, the semantic similarity between *Synsets* can be calculated. A word may have multiple semantics and parts of speech corresponding to different *Synsets*. Therefore, the two words have different semantic similarities in different *Synsets* (see Table 2). We have to choose one of them as the semantic similarity between two words. Some of the existing methods choose the maximum value [24, 25]. This is the basis for calculating the semantic similarity between documents.

4. Study Methods

This section consists of three parts. Section 4.1 introduces the method of calculating the semantic similarity between two service documents. Section 4.2 introduces the method of using semantic similarity to calculate the vector corresponding to the service document. Section 4.3 introduces how to select the appropriate algorithm to cluster the vectors.

The main study methods of this paper are based on (1) obtaining preprocessed documents (PD) through tags and WordNet, (2) calculating the semantic similarity between PDs and then obtaining the semantic distance matrix, (3) using the multidimensional scaling to analyze the semantic distance matrix to obtain the semantic distance vector (SDV) corresponding to each web service, and (4) using the K-means algorithm to cluster the SDVs to achieve clustering of web services (see Figure 3). The multidimensional scaling is used to translate "information about the pairwise "distances" among a set of n objects or individuals" into a configuration of n points mapped into an abstract Cartesian space [26].

4.1. Calculate Semantic Similarity. The basis of calculating document semantic similarity is to calculate the semantic similarity between words. We enumerate all the semantic similarities of two words in different *Synsets* and select the largest as the semantic similarity of the words [24, 25].

Before calculating the semantic similarity of documents, preprocessing is required. General preprocessing methods include removing punctuation and stop words. This paper considers the particularity of Web service description documents. Except for stop words, there are many words that have nothing to do with document semantics. "Tag" is the auxiliary information given by web service developers according to the research results of Jingli et al. In [27], using tags can filter out the words that are not related to the topic; according to the research results of Shi et al. [28], the more tags two web services have duplicates, the more likely they are to belong to the same category. Therefore, in the process of document preprocessing, we keep words that are semantically similar to tags, thereby removing words that have nothing to do with the subject of the document. We use D to represent the service description document, T to represent the document tag collection, and PD to represent the preprocessed document (see Algorithm 1).

Regarding the threshold α, since the semantic similarity calculation result of WordNet is between 0 and 1, we adopt an intermediate value strategy and take α as 0.5.

The semantic similarity between the two PDs is determined by the words in the PD (see Figure 4). We can calculate the maximum semantic similarity between each word and all the words on the opposite side. The semantic similarity between two PDs is divided by the sum of length after the similarity is accumulated, which can ensure the symmetry (see Algorithm 2).

The semantic similarity calculated by WordNet is between 0 and 1.

$$
\begin{aligned}
&0 < \text{Wordnet.similary}\left(A_i, B_j\right) < 1,\\
&\text{SUM}\left(\text{PD1}\right) = \sum_{i=1}^{m} \text{Max}\left(\text{Wordnet.similary}\left(A_i, B_j\right)\right) \quad (1 < j < n),\\
&\text{SUM}\left(\text{PD2}\right) = \sum_{j=1}^{n} \text{Max}\left(\text{Wordnet.similary}\left(B_j, A_i\right)\right) \quad (1 < i < m).
\end{aligned}
\tag{1}
$$

So, we can get

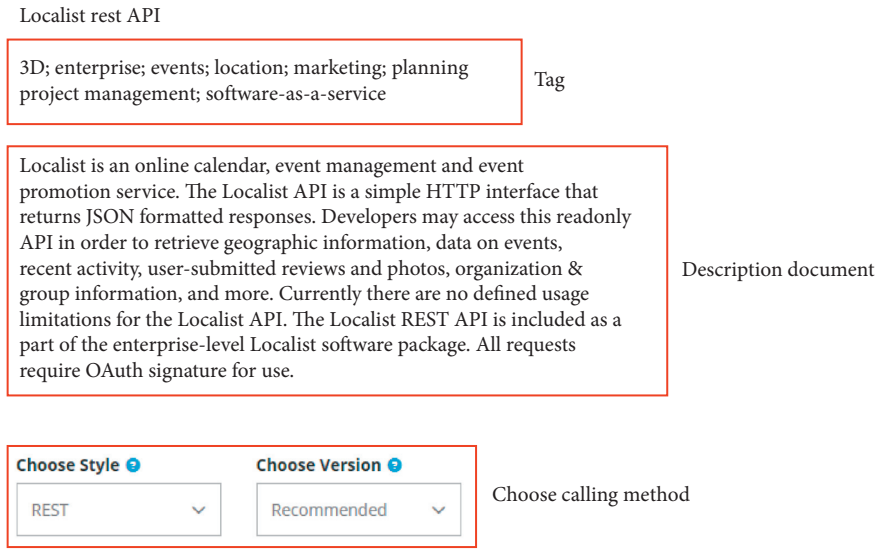


Figure 1: A web service with the category "Calendars" on ProgrammableWeb [40].

Table 1: Different meanings in different *Synsets* of the word "people."

Synset	Meaning
people.n.01	(Plural) any group of human beings (men or women or children) collectively
people.n.02	The body of citizens of a state or country facilities for research and teaching
people.n.03	Members of a family line
People.n.04	The common people generally

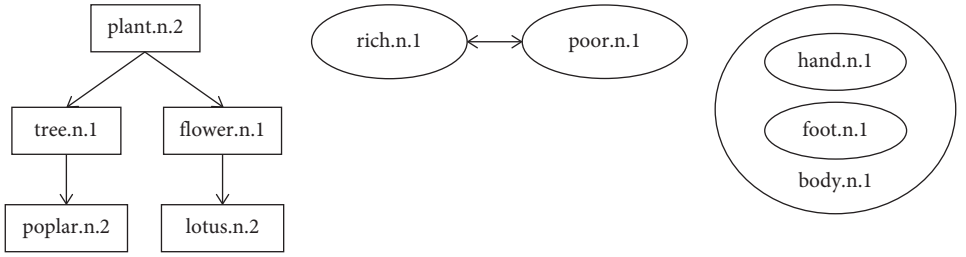


Figure 2: Three relationships between *Synset*: (a) upper and lower relation; (b) antisense relation; (c) whole and part relation.

Table 2: Semantic similarity between "people" and "citizenry" in different *Synsets*.

	citizenry.n.01
people.n.01	0.33333
people.n.02	1
people.n.03	0.1250
People.n.04	0.33333

$$0 < \mathrm{SUM}(\mathrm{PD1}) < m,$$
$$0 < \mathrm{SUM}(\mathrm{PD2}) < n, \tag{2}$$
$$0 < \mathrm{sim}(\mathrm{PD1}, \mathrm{PD2}) = \frac{\mathrm{SUM}(\mathrm{PD1}) + \mathrm{SUM}(\mathrm{PD2})}{m+n} < 1.$$

We assume that the number of web service description documents is n. Through this algorithm, we can get a

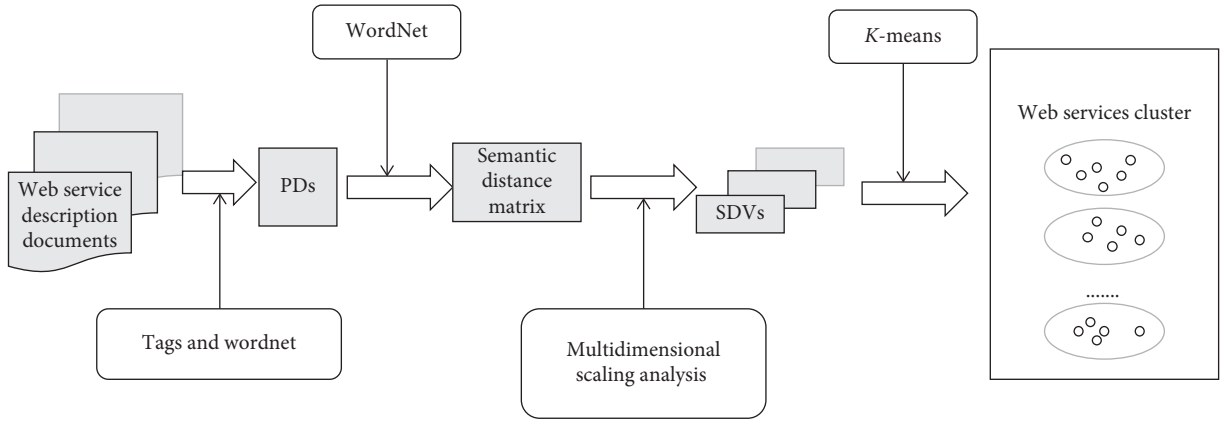


Figure 3: The framework of our web services clustering method.

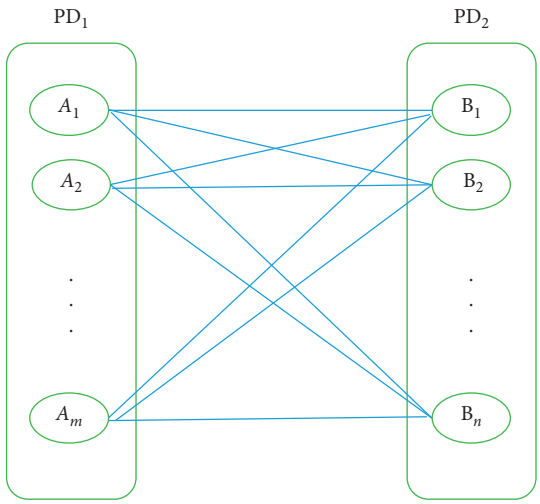


Figure 4: The semantic similarity between two PDs. The length is m and n, respectively.

semantic similarity matrix called $\mathrm{SIM} = (\mathrm{sim}_{ij})_{n\times n}$. The matrix elements are between 0 and 1, the larger the element value, the higher the semantic similarity. The sim_{ij} represents the semantic similarity between the i-th and j-th documents. Obviously, all diagonal elements are 1.

4.2. Multidimensional Scaling Analysis. The problem solved by the multidimensional scaling method is as follows. When the similarity (or distance) of each pair of n objects is given, the representation of these objects in multidimensional space is determined, and the original similarity (or distance) is expressed as much as possible. In other words, two semantic similar web services are represented by two points close to each other in multidimensional space, which creates conditions for clustering [29]. We first introduce data concepts related to multidimensional scaling.

4.2.1. Similar Data and Distance Data

Similar Data. This is the data representing the similarity of two objects. The larger the value is, the higher the similarity is. "Semantic similarity" in the previous article is the *similar data*.

Distance Data. This is contrary to similar data. The larger the value is, the lower the similarity is.

Only the distance data can be directly used for multidimensional scaling analysis [29].

4.2.2. Distance Matrix. A matrix $\mathrm{DIS} = (\mathrm{dis}_{ij})_{n\times n}$ of order $n \times n$, dis_{ij} is the distance between the i-th object and the j-th object if the following condition is met:

$$\begin{aligned} &\mathrm{DIS} = \mathrm{DIS}^T, \\ &\mathrm{dis}_{ij} \geq 0,\ \mathrm{dis}_{ii} = 0, \quad i, j = 1, 2, 3, \ldots, n. \end{aligned} \tag{3}$$

Then, the matrix DIS is a distance matrix.

If there is a positive integer r and there are n points in R^r, $X_1, X_2, \ldots, X_n$, such that

$$\mathrm{dis}_{ij}^2 = \left(X_i - X_j\right)^T \left(X_i - X_j\right), \tag{4}$$

then DIS is called the Euclidean distance matrix [30]. In fact, there is a simpler way to determine whether the distance matrix is a Euclidean distance matrix, which we will introduce in later chapters.

4.2.3. Similarity Coefficient Matrix. A matrix $C = (c_{ij})_{n\times n}$ of order $n \times n$, c_{ij} is the similarity coefficient between the i-th object and the j-th object if the following condition is met:

$$\begin{aligned} &C = C^T, \\ &c_{ij} \leq c_{ii}, \quad i, j = 1, 2, 3, \ldots, n. \end{aligned} \tag{5}$$

Then, matrix C is a similarity coefficient matrix.

If the data are not a distance matrix, it must be transformed into a distance matrix by a certain method in order to carry out multidimensional scaling analysis.

Therefore, the semantic similarity matrix SIM is not suitable for multidimensional scaling analysis. We need to translate semantic similarity into "semantic distance" through inversion. We define "semantic distance" as a value from two service description documents, between 0 and 1.

The smaller the semantic distance value, the higher the semantic similarity. The semantic distance matrix is $\mathrm{DIS} = (\mathrm{dis}_{ij})_{n\times n}$. We need to use appropriate functions to reverse the semantic similarity. There is a classic transformation function (see equation (6)) [31–33]. However, from the experimental point of view, the effect of this function is not satisfactory.

$$\mathrm{dis}_{ij} = \left(c_{ii} + c_{jj} - 2c_{ij}\right)^{1/2}. \tag{6}$$

The sigmoid function is a commonly used activation function in neural networks [34]. This function expression is shown in the following equation:

$$S(x) = \frac{1}{1 + e^{-x}}. \tag{7}$$

The sigmoid function is an increasing function, and it cannot activate the data between 0 and 1, so we need to deform it. We use this function (see equation (8)) to reverse the data.

$$\mathrm{NS}(x) = \frac{1}{1 + e^{\sigma(x-\mu)}}. \tag{8}$$

μ and σ are adjustment coefficients. To ensure that NS is a minus function, μ and σ should be positive real numbers. We adjust μ and σ for many times through experiments and determine that when $\sigma = 20$ and $\mu = 0.3$, we can get better results.

So, we can calculate DIS by the following equation:

$$\mathrm{dis}_{ij} = \begin{cases} \mathrm{NS}\left(\mathrm{sim}_{ij}\right), & i \neq j \\ 0, & i = j \end{cases} \tag{9}$$

Let n points in r-dimensional space be expressed as $X_1, X_2, \ldots, X_n$ and expressed by matrix as $X = (X_1, X_2, \ldots, X_n)^T$. If the corresponding point of the i-th web service description document is X_i, then the coordinate of X_i is marked as follows:

$$X_i = \left(X_{i1}, X_{i2}, \ldots, X_{ir}\right) = \mathrm{SDV}_i. \tag{10}$$

The purpose of multidimensional scaling analysis is to calculate X. We call X a fitting composition of the semantic distance matrix DIS.

Let $B = (b_{ij})_{n\times n}$, where B is called the central inner product matrix of X, and the construction of matrix B is the premise of multidimensional scaling analysis [29]. Let us first construct matrix $A = (a_{ij})_{n\times n}$ according to the following equation:

$$a_{ij} = -\frac{1}{2}\mathrm{dis}_{ij}^2. \tag{11}$$

Next, the matrix $H = (h_{ij})_{n\times n}$ is constructed according to equation (12). In equation (12), I_n is an identity matrix of order n, E_n is a square matrix of order n, and any element of matrix E_n is 1.

$$H = I_n - \frac{1}{n}E_n. \tag{12}$$

Finally, matrix B is constructed as follows:

$$B = HAH. \tag{13}$$

We calculate the n eigenvalues of B and arrange them to obtain

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n. \tag{14}$$

The eigenvectors corresponding to the n eigenvalues are

$$e_1, e_2, e_3, \ldots, e_n. \tag{15}$$

The sufficient and necessary condition for the semantic distance matrix DIS to be Euclidean distance matrix is $|B| \geq 0$ [29]. We will discuss two cases of $|B|$.

(1) When $|B| \geq 0$, DIS is a Euclidean distance matrix, and all eigenvalues are nonnegative:

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n \geq 0. \tag{16}$$

The dimension of coordinate Xi is r. We need to construct X by using the eigenvector corresponding to r maximum eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_r$. r can be determined by accumulating the eigenvalues and calculating the proportion of the accumulated sum to the sum of all eigenvalues.

$$\frac{\lambda_1 + \lambda_2 + \lambda_3 + \ldots + \lambda_r}{\lambda_1 + \lambda_2 + \lambda_3 + \ldots + \lambda_n} \geq \alpha. \tag{17}$$

α is the threshold given in advance, generally 80% [29]. Then, $e_1, e_2, e_3, \ldots, e_r$ are selected to construct X.

(2) When $|B| < 0$, DIS is a non-Euclidean distance matrix. And there are negative eigenvalues.

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_s \geq 0 > \lambda_{s+1} \geq \ldots \geq \lambda_n. \tag{18}$$

r can be determined by accumulating the eigenvalues and calculating the proportion of the accumulated sum to the sum of absolute values of all eigenvalues.

$$\frac{\lambda_1 + \lambda_2 + \lambda_3 + \ldots + \lambda_r}{\left|\lambda_{1|}\right| + \left|\lambda_2\right| + \left|\lambda_3\right| + \ldots + \left|\lambda_n\right|} \geq \alpha. \tag{19}$$

α is the threshold given in advance, generally 80%. Then, $e_1, e_2, e_3, \ldots, e_r$ are selected to construct X.

Next, X is calculated as follows:

$$X = \left(\sqrt{\lambda_1}\, e_1, \sqrt{\lambda_2}\, e_2, \ldots, \sqrt{\lambda_r}\, e_r\right) = \left(x_{ij}\right)_{n\times r}. \tag{20}$$

Each line in X corresponds to a web service description document, and the i-th line is Xi. Next, we need to cluster $X_i\,(1 \leq i \leq n)$.

4.3. Clustering Algorithm. The existing clustering methods are mainly divided into: layering, partitioning, density-based, model-based, grid-based, and soft computing methods [35]. We project the SDV into a two-dimensional space through PCA [36]. We analyzed the distribution of SDV projections and believed that the partitioning

clustering method [37] is suitable for processing our data. We compared each partitioning clustering method through experiments and chose the *K*-means algorithm. The *K*-means algorithm is a classic unsupervised learning clustering method, which is used in this paper for service clustering. [38, 39].

At this point, our research methods are all introduced.

5. Experimental Results and Analyses

5.1. Experimental Data. Our experimental data are real data crawled from the ProgrammableWeb. As of October 3, 2020, there were 21956 web services on ProgrammableWeb, totaling 425 categories [40]. The number of web services covered by different topics varies greatly. For example, there are 1020 web services in the category *Financial* and only one web service in the category IDE. The number is too unbalanced, which seriously affects the clustering effect. For this experiment, we select the categories that contain more than 400 web services. There are 11 categories ($CG = \{CG_1, CG_2, \ldots, CG_{11}\}$), including 6533 web services (see Table 3). These classifications are completed by the developers who publish these web services and are generally considered to be accurate.

5.2. Evaluating Indicator. We use three indexes to evaluate the clustering effect, which are precision, recall, and F-measure. We cluster 6533 web services into 11 clusters, which are expressed as $NG = \{NG_1, NG_2, \ldots, NG_{11}\}$. The three indexes are defined as follows:

$$\text{precision}(NG_i) = \frac{|CG_i \cap NG_i|}{|NG_i|},$$

$$\text{recall}(NG_i) = \frac{|CG_i \cap NG_i|}{|CG_i|},$$

$$F - \text{measure}(NG_i) = \frac{2 * \text{Precision}(NG_i) * \text{Recall}(NG_i)}{\text{Precision}(NG_i) + \text{Recall}(NG_i)}. \tag{21}$$

5.3. Comparison Method. Our method is compared with these five methods. The introduction is as follows:

(1) *TF-IDF + K* [15]. Keywords are extracted by word frequency and inverse document word frequency, and document-keyword vectors are constructed with keywords. *K*-means is used to cluster the document-keyword vectors.

(2) *LDA + K* [16]. We use latent Dirichlet allocation to model the documents and then get the topic-word matrix and document-topic vectors. *K*-means is used to cluster document-topic vectors.

(3) *Doc2Vec + K* [19]. We use the Doc2Vec model to train the documents and convert the documents into vectors. *K*-means is used to cluster document vectors.

(4) *RCNN + LDA + K* [20]. First train the LDA model to obtain the probability-topic distribution of each document and then train the RCNN network to obtain a fitting model from each service document to the probability-topic distribution. In this process, the feature vector of each service document can be obtained. Finally, cluster the document-feature vectors by *K*-means.

(5) *CMD + CT + K*. The classical transformation function is used to process similar data, and then multidimensional scaling analysis is carried out (see equation (6)). Finally, *K*-means clustering is used. We want to demonstrate the effectiveness of our new transformation method through this experiment.

(6) *WMS*. The new method proposed in this paper.

In order to compare the performance of the methods more objectively, we use Algorithm 1 to preprocess documents for all six methods, .

5.4. Comparison of Experimental Results.

(1) Algorithm implementation.

The distance matrix DIS with a dimension of 6533 can be obtained by processing the experimental data using the method designed above. We need to determine if the DIS is a Euclidean distance matrix. We calculated the eigenvalues and eigenvectors of the DIS and got 6533 eigenvalues (see Figure 5):

A total of 2032 eigenvalues are negative, so DIS is a non-Euclidean distance matrix. Let us take $\alpha = 80\%$, and when $r = 50$, equation (19) is satisfied, so we take the vector dimension as 50.

(2) Precision comparison of 6 methods on 5 categories (see Figure 6).

(3) Recall comparison of 6 methods on 5 categories (see Figure 7).

(4) F-measure comparison of 6 methods on 5 categories (see Figure 8).

(5) The average precision, recall, and F-measure of the 6 methods on 11 categories (see Table 4).

5.5. Result Analysis. From the experimental results, the TF-IDF + *K* method is the worst because the service description document is too short to extract keywords although the clustering effect is improved by adding context information. It should be noted that the LDA + *K* method, the Doc2Vec + *K* method, and the RCNN + LDA + *K* method contain a large number of random processes, resulting in different operation results in each operation. In contrast, the WMS method proposed in this paper not only has stable results but also has the best clustering effect. From the results, the clustering effect of the Doc2Vec + *K* method and the RCNN + LDA + *K* method is poor. We believe that these two methods rely on the continuity of the document, but our preprocessing (see Algorithm 1) destroys the continuity of the document. In contrast, the LDA + *K* method and the

```
Input: D, T, α
Output: PD (PD is initialized to empty)
Begin:
FOR each tag in T do:
  flag ⟵ 0;
  FOR each word in D do:
    IF WordNet.similarity (tag, word) >α do:
      PD.add (word);
      flag ⟵ 1;
    ENDIF
  ENDFOR
  IF flag = 0 do:
    PD.add (tag);
  ENDIF
ENDFOR
RETURN PD;
END
```

Algorithm 1: Preprocessing of a service description document.

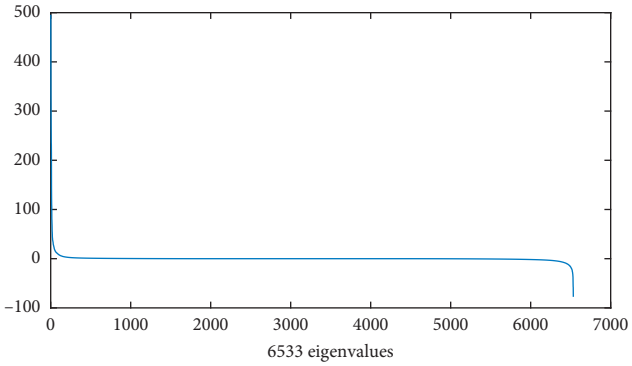


Figure 5: 6533 eigenvalues of distance matrix DIS.

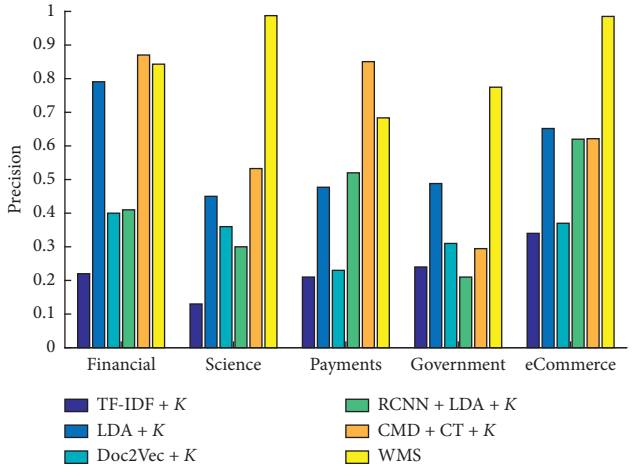


Figure 6: Precision of 6 methods on top 5 categories.

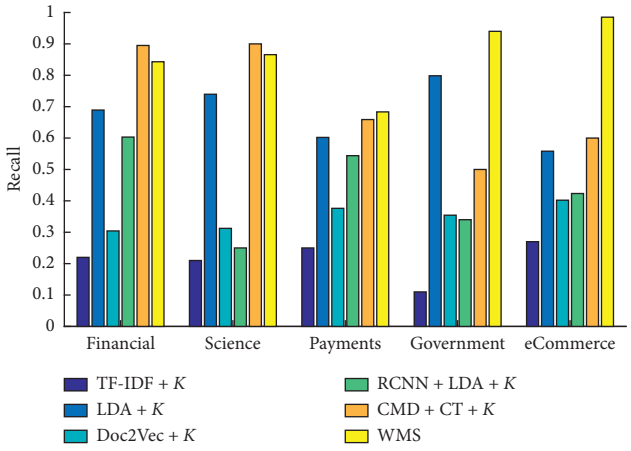


Figure 7: Recall of 6 methods on 5 categories.

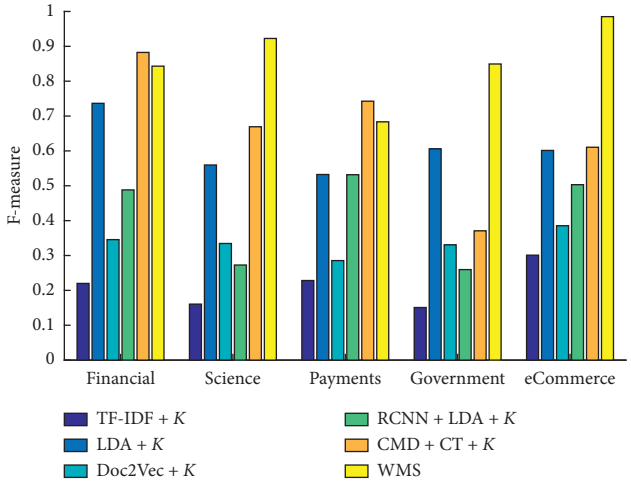


Figure 8: F-measure of 6 methods on 5 categories.

```
Input: PD1 (the length is m), PD2 (the length is n)
Output: sim (the semantic similarity between two PDs)
Begin:
SUM ⟵ 0;
FOR i ⟵ 1 to m do:
   MAX ⟵ 0;
   FOR j ⟵ 1 to n do:
      IF WordNet.similarity (A_i, B_j) >MAX do:
         MAX ⟵ WordNet.similarity (A_i, B_j);
      ENDIF
   ENDFOR
   SUM ⟵ SUM + MAX;
ENDFOR
FOR j ⟵ 1 to n do:
   MAX ⟵ 0;
   FOR i ⟵ 1 to m do:
      IF WordNet.similarity (A_i, B_j) >MAX do:
         MAX ⟵ WordNet.similarity (A_i, B_j);
      ENDIF
   ENDFOR
   SUM ⟵ SUM + MAX;
ENDFOR
sim ⟵ SUM/(m + n);
RETURN sim;
END
```

Algorithm 2: The process of calculating the semantic similarity between two PDs.

WMS method do not have any requirements for document continuity, so the clustering effect is better. And we improved the method of transposing data in multidimensional scaling analysis. Experiments prove that our improvement is effective.

5.6. Selection of Clustering Algorithm. We project the SDV into a two-dimensional space through PCA (see Figure 9).

We can see from Figure 9 that the clusters of SDV data are roughly distributed around a certain center in an elliptical shape, and some clusters are more fused. The partitioning clustering method is suitable for processing such data [39]. And our data belong to numerical type data. There are three typical partitioning clustering methods suitable for processing numerical type data: *K*-means, *K*-medoids [41], and Clustering for Large Application (CLARA) [42]. We compared the average precision, average recall, and average F-measure of the three methods (see Table 5). It is finally determined that *K*-means has the best clustering effect.

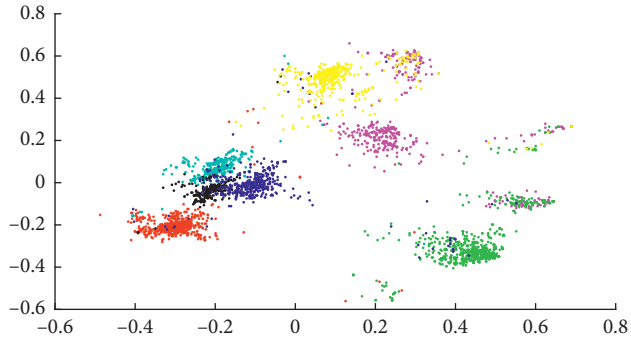


Figure 9: SDVs projection in two-dimensional space.

5.7. Supplementary Notes. Here, we show how to determine $\mu = 0.3$ and $\sigma = 20$ in equation (8). The symmetry center of sigmoid function is $(\mu, 0)$, so μ plays the role of data segmentation. We count the distribution of elements in the matrix SIM (see Figure 10).

Table 3: The distribution of web services in top 18 categories.

Category	Number
Financial	1020
Tools	856
Payments	657
Messaging	650
E-commerce	627
Enterprise	508
Social	497
Mapping	470
Science	434
Government	412
Security	402

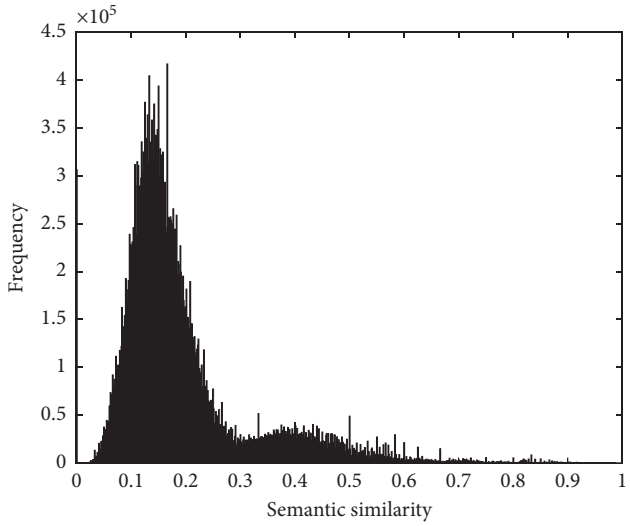


Figure 10: The distribution of elements in the matrix SIM.

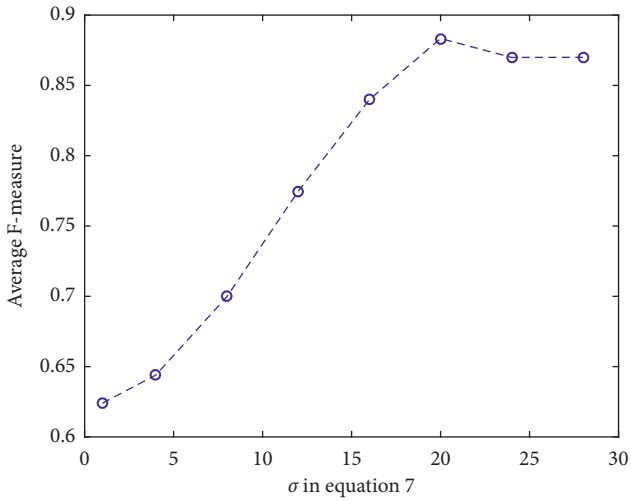


Figure 11: The changes of average F-measure by adjusting σ

It can be found that 0.3 is the segmentation point of frequency, so $\mu = 0.3$. After that, determine $\mu = 0.3$ and record the changes of average F-measure by adjusting σ in equation (8) (see Figure 11).

Table 4: Comparison of average precision, recall, and F-measure of six methods.

Method	Precision	Recall	F-measure
TF-IDF + K	0.1936	0.2010	0.1986
LDA + K	0.5827	0.6461	0.5994
Doc2Vec + K	0.3502	0.3096	0.3203
RCNN + LDA + K	0.5231	0.5763	0.5496
CMD + CT + K	0.7816	0.8311	0.7967
WMS	0.8935	0.8734	0.8795

Table 5: Comparison of average precision, recall, and F-measure of three methods.

Method	Precision	Recall	F-measure
K-medoids	0.8087	0.8650	0.8206
CLARA	0.8138	0.8705	0.8269
K-means	0.8935	0.8734	0.8795

As can be seen from Figure 11, when $\sigma = 20$, the average F-measure has good result.

6. Conclusions

In this paper, we propose a web service clustering method based on semantic similarity and multidimensional scaling analysis. We first used WordNet to calculate the semantic similarity between documents and then obtained the semantic distance matrix. Then, we used multidimensional scaling analysis to get the SDVs. Finally, we used the K-means algorithm to cluster the SDVs. Most of the existing methods vectorize documents by extracting document features. We have proposed a new idea to vectorize documents by comparing the differences between documents. The improvement of the vectorization method leads to the improvement of the clustering effect. Multidimensional scaling analysis is the core of our method. The experimental results show that our method is better than existing methods in precision, recall, and F-measure. And our method is more deterministic than the method based on deep neural network and LDA. And we improved the method of transposing data in multidimensional scaling analysis. Experiments prove that our improvement is effective.

We believe that our method has a major flaw; our algorithm relies on tags and is less robust. For future work, we will improve Algorithm 2 to get rid of the dependence on tags. In addition, service clustering cannot be directly useful to users. For future work, we will use the service clustering method in this article as a basis to improve service composition, service discovery, and other web service tasks.

Data Availability

The data used to support the results of this study are obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (61572195) and the special fund of Shanghai Economic and Information Commission (sheitc160306).

References

[1] J. Pasley, "How BPEL and SOA are changing web services development," *IEEE Internet Computing*, vol. 9, no. 3, pp. 60–67, 2005.

[2] N. Zhang, J. Wang, K. He, Z. Li, and Y. Huang, "Mining and clustering service goals for RESTful service discovery," *Knowledge and Information Systems*, vol. 58, no. 3, pp. 669–700, 2019.

[3] I. Lizarralde, C. Mateos, A. Zunino, T. A. Majchrzak, and T.-M. Grønli, "Discovering web services in social web service repositories using deep variational autoencoders," *Information Processing & Management*, vol. 57, no. 4, Article ID 102231, 2020.

[4] C. Sun, L. Lv, G. Tian, Q. Wang, X. Zhang, and L. Guo, "Leverage label and word embedding for semantic sparse web service discovery," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–8, Article ID 5670215, 2020.

[5] B. Xia, Y. Fan, W. Tan et al., "Categoryaware API clustering and distributed recommendation for automatic mashup creation," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 674–687, 2014.

[6] B. Cao, X. Liu, M. M. Rahman et al., "Integrated content and network-based service clustering and web apis recommendation for mashup development," *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 99–113, 2017.

[7] M. Shi, J. Liu, D. Zhou et al., "A probabilistic topic model for mashup tag recommendation," in *Proceedings of IEEE International Conference on Web Services (ICWS)*, pp. 444–451, San Francisco, CA, USA, June 2016.

[8] B. T. G. S. Kumara, I. Paik, W. Chen, and K. H. Ryu, "Web service clustering using a hybrid term-similarity measure with ontology learning," *International Journal of Web Services Research*, vol. 11, no. 2, pp. 24–45, 2014.

[9] R. A. H. M. Rupasingha, I. Paik, and B. T. G. S. Kumara, "Specificity-aware ontology generation for improving web service clustering," *IEICE Transactions on Information and Systems*, vol. E101-D, no. 8, pp. 2035–2043, Aug. 2018.

[10] A. Konduri, "Clustering of web services based on semantic similarity," MS thesis, University of Akron, Akron, OH, USA, 2008.

[11] B. T. G. S. Kumara, I. Paik, and Y. Yaguchi, "Context-aware web service clustering and visualization," *International Journal of Web Services Research*, vol. 17, no. 4, pp. 32–54, 2020.

[12] T. Liang, Y. Chen, W. Gao, M. Chen, M. Zheng, and J. Wu, "Exploiting user tagging for web service Co-clustering," *IEEE Access*, vol. 7, pp. 168981–168993, 2019.

[13] Y. Gu, H. Cai, C. Xie, L. Jiang, Y. Gu, and A. Liu, "Utilizing semantic information from linked open data in web service clustering," in *Proceedings of 2016 International Conference on Progress in Informatics and Computing (PIC)*, pp. 654–658, Shanghai, China, December 2016.

[14] N. Agarwal, G. Sikka, and L. K. Awasthi, "Enhancing web service clustering using Length Feature Weight Method for service description document vector space representation," *Expert Systems with Applications*, vol. 161, Article ID 113682, 2020.

[15] A. Muath and D. Inkpen, "Clustering the topics using TF-IDF for model fusion," in *Proceedings of Phd Workshop on Information & Knowledge Management ACM*, Napa Valley, CF, USA, October 2008.

[16] C. Lin, Y. He, C. Pedrinaci, and J. Domingue, "Feature LDA: a supervised topic model for automatic detection of web API documentations from the web," in *Proceedings of International Semantic Web Conference Springer*, Berlin, Heidelberg, November 2012.

[17] M. Shi, J. Liu, D. Zhou, M. Tang, and B. Cao, "WE-LDA: a word embeddings augmented LDA model for web services clustering," in *Proceedings of 2017 IEEE International Conference on Web Services (ICWS)*, pp. 9–16, Honolulu, HI, USA, June 2017.

[18] Y. Zhao, K. He, and Y. Qiao, "ST-LDA: high quality similar words augmented LDA for service clustering," in *Proceedings of International Conference on Algorithms and Architectures for Parallel Processing*, pp. 46–59, Springer, Guangzhou, China, November 2018.

[19] X. Zhang, J. Liu, B. Cao, Q. Xiao, and Y. Wen, "Web service recommendation via combining Doc2Vec-based functionality clustering and DeepFM-based score prediction," in *Proceedings of 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pp. 509–516, Melbourne, Australia, December 2018.

[20] G. Zou, Z. Qin, Q. He, P. Wang, B. Zhang, and Y. Gan, "DeepWSC: a novel framework with deep neural network for web service clustering," in *Proceedings of 2019 IEEE International Conference on Web Services (ICWS)*, pp. 434–436, Milan, Italy, July 2019.

[21] G. Zou, Z. Qin, Q. He, P. Wang, B. Zhang, and Y. Gan, "DeepWSC: clustering web services via integrating service composability into deep semantic features," *IEEE Transactions on Services Computing*, vol. 9, 2020.

[22] "ProgrammableWeb" from Wikipedia, 2020, https://en.wikipedia.org/wiki/ProgrammableWeb.

[23] "WordNet" from Wikipedia, 2020, https://en.wikipedia.org/wiki/WordNet.

[24] J. Gonzalo, M. F. Verdejo, I. Chugur, and J. Cigarran, "Indexing with WordNet synsets can improve text retrieval," arXiv preprint cmp-lg/9808002, 1998.

[25] C. Strapparava and A. Valitutti, "Wordnet affect: an affective extension of wordnet," *Lrec*, vol. 4, pp. 1083–1086, 2004.

[26] A. Mead, "Review of the development of multidimensional scaling methods," *The Statistician*, vol. 41, no. 1, pp. 27–39, 1992.

[27] Z. Jingli, N. Xuejun, Q. Leihua et al., "Web clustering based on tag set similarity," *Journal of Computers*, vol. 6, no. 1, pp. 59–66, 2011.

[28] M. Shi, J. Liu, B.-Q. Cao, Y. Wen, and X. Zhang, "A prior knowledge based approach to improving accuracy of web services clustering," in *Proceedings of 2018 IEEE International Conference on Services Computing (SCC)*, IEEE, San Francisco, CA, USA, July 2018.

[29] M. A. A. Cox and T. F. Cox, "Multidimensional scaling," *Handbook of Data Visualization*, Springer, Berlin, Heidelberg, pp. 315–347, 2008.

[30] J. C. Gower, "Properties of Euclidean and non-Euclidean distance matrices," *Linear Algebra and Its Applications*, vol. 67, pp. 81–97, 1985.
[31] W. Härdle and L. Simar, "Multidimensional scaling," in *Applied Multivariate Statistical Analysis*Springer, Berlin, Heidelberg, 2003.
[32] C. K. I. Williams, "On a connection between kernel PCA and metric multidimensional scaling," *Machine Learning*, vol. 46, no. 1/3, pp. 11–19, 2002.
[33] W. K. Härdle and L. Simar, "Multidimensional scaling," in *Applied Multivariate Statistical Analysis*Springer, Berlin, Heidelberg, 2019.
[34] X. Yin, J. A. N. Goudriaan, E. A. Lantinga et al., "A flexible sigmoid function of determinate growth," *Annals of Botany*, vol. 91, no. 3, pp. 361–371, 2003.
[35] L. Rokach, "A survey of clustering algorithms," *Data Mining and Knowledge Discovery Handbook*, Springer, Boston, MA, USA, pp. 269–298, 2009.
[36] S. Wold, E. Kim, and G. Paul, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.
[37] J. Swarndeep Saket and P. Sharnil, "An overview of partitioning algorithms in clustering techniques," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 5, no. 6, pp. 1943–1946, 2016.
[38] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: a k-means clustering algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979.
[39] M. Agarwal, R. Jaiswal, and A. Pal, "k-means++ under approximation stability," *Theoretical Computer Science*, vol. 588, pp. 37–51, 2015.
[40] ProgrammableWeb, 2020, https://www.programmableweb.com/.
[41] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.
[42] L. Kaufman and P. J. Rousseeuw, —*Finding Groups in Data*‖, John Wiley, New York, NY, USA, 1990.

Hindawi
Scientific Programming
Volume 2021, Article ID 6619712, 11 pages
https://doi.org/10.1155/2021/6619712

Research Article

A Hotspot Information Extraction Hybrid Solution of Online Posts' Textual Data

HuiRu Cao,[1] Xiaomin Li,[2] Songyao Lian,[3] and Choujun Zhan[4]

[1]*Department of Information Engineering, Guangzhou Institute of Technology, Guangzhou 510725, China*
[2]*College of Mechanical and Electrical Engineering, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China*
[3]*Nanfang College, Sun Yat-sen University, Guangzhou 510970, China*
[4]*School of Computer, South China Normal University, Guangzhou 510631, China*

Correspondence should be addressed to Xiaomin Li; lixiaomin@zhku.edu.cn

Received 7 November 2020; Revised 28 March 2021; Accepted 7 April 2021; Published 15 April 2021

Academic Editor: J. R. Méndez

Online posts have gradually become a major carrier of network public opinion in social media, and the social network hotspots are the important basis for the study of network public opinion. Therefore, it is significant to extract hotspots for monitoring Internet public opinion from online posts textual big data. However, the current hotspot extraction methods are focused on the users' features that are based on textual big data with spam and low-quality content. Meanwhile, these methods seldomly consider the time span of posts and the popularity of users. Accordingly, this article presents a hotspots information extraction hybrid solution of online posts' textual data. Firstly, a filtering strategy to obtain more high-quality textual data is designed. Secondly, the topic hot degree is presented by considering the average number of replies and the popularity of the participant. Thirdly, an improved co-word analysis technology is used to search the same topic posts and Bisecting k-means clustering algorithm using repliers' popularity and key posts are designed for studying and monitoring the hotspots of online posts in a valid big data environment. Finally, the proposed algorithms are verified in experiments by extracting the hotspots of online posts from the dataset. The results show that the data filtering strategy can help to obtain more valuable information and decrease the computing time. The results also demonstrate that the proposed solution can help to obtain hotspots comparing the traditional methods, and the hot degree can reflect the trend of the online post by comparing the traditional methods.

1. Introduction

With the rapid development of mobile communications and networks, the Internet increasingly integrates into our life. It is reported that there are now more than 4 billion Internet users around the world. Most Internet users spend an average of six hours surfing the Internet, and 3 billion people now use social media, such as Twitter, blogs, Bulletin Board System (BBS), and podcasts [1, 2]. It is known that online posts have gradually become an important tool in social media for the exchange of information. An increasing amount of public opinion is now spread by social media, especially through BBS [3–5]. Since hotspots directly reflect public opinion, studying and monitoring the hotspots of social media becomes more important for public affairs.

Social media has become one of the most important and popular carriers and distributors of the current online public opinion [6, 7]. Compared to the traditional public opinion channels, online posts have some unique features, such as a wider audience range, greater influence, faster propagation speed, and large amount of data [8–10]. For obtaining and monitoring public opinion hotspots, an increased number of studies focus on this field from different perspectives. In general, the current studies mainly use natural language processing, data mining technologies, machine learning, and other methods to monitor hotspots and explore propagation [11–14].

Currently, text data is still an important medium for information dissemination on social networks [15, 16]. To study complex dynamics in social networks, the extraction of hotspots from massive textual data becomes one of the important steps. On the one hand, the current hotspots' extraction methods are simple to collect the user-related feature and mostly based on textual big data with spam, irrelevant, and low-quality content. In social media, there are many spam information [17–19], such as paid posters and fake replies, as shown in Figure 1. Advertising posts and replies are a good example in BBS. Such corresponding users' featured information based on invalid or incomplete data can be very different from real one, especially for hotspots and public opinions. On the other hand, the main methods seldomly take into account the time span of posts and popularity of repliers. Firstly, time span of posts is a significant factor in hotspots extraction, as hotspots of social networks are the collective action of users in a short time (for example, a collective response to an event in BBS). Secondly, it is reported that popularity users (repliers and main posters) play a significant role in Internet public opinion [20]. However, few studies address these problems. Due to the complexity and features of social media such as BBS, monitoring of public opinion hotspots still faces the following challenges:

(1) How to obtain more valuable data by filtering a large amount of spam textual data

(2) How to find the key posts according to the association among multiple posts for the same topic

(3) How to search real hotspots by considering the valuable repliers and key posts

Accordingly, for solving the problems, a hotspots' information extraction hybrid solution of online posts' textual data is proposed based on the feature of users in social networks. The solution contains three main steps. Firstly, a textual data filtering strategy is used to obtain a more valid dataset. An improved co-word analysis technology is used to search the same topic posts. Secondly, bisecting k-means clustering algorithm based on poster popularity and key posts are proposed to obtain the hotspots of online posts. Then, the hot degree is proposed to search the real hotspots. The proposed methods are implemented in a real experiment, where the results demonstrate the effectiveness of the solution.

The rest of the paper is structured as follows. Section 2 discusses related studies on the Internet public opinion and current challenges. Section 3 introduces hotspot monitoring and public opinion communication characteristics. Section 4 introduces the cluster hotspot monitoring based on PR values and bisecting k-mean algorithms. Section 4 presents the results of an experiment using the proposed methods and our dataset. Section 5 concludes and discusses the paper.

2. Related Work

In this section, we present existing studies on the public opinion analysis of BBS and monitoring hotspots. These studies are used as a basis for our work. We review related research from two aspects: network public opinion and hotspots.

2.1. Public Opinion. In [21], the natural language processing and machine learning techniques are used to interpret sentimental tendencies related to users' opinions and predict real events. In [22], a public opinion dynamics model for an online-offline social network context is provided and conditions to form a consensus in the proposed model are analyzed. In [23], the authors propose a method to recognize network public opinion leaders by using Markov logic networks, and a recognition system is designed and implemented. In [24], a cross-network public opinion spreading model is created in a combined social network environment. Two network nodes are assumed in this paper. In [25], the author constructs a dictionary monitoring sentiment computing model using text words and labels as the input parameters. In [26], a new method is provided for sentiment computing for news and events by constructing a word emotion association network. The authors provide a word emotion computation method to obtain initial words. These studies mostly focus on public opinion based on the assumption that the dataset is always valid.

2.2. Hotspots. Zhao et al. [27] present a Social Sentiment Sensor (SSS) system on Sina Weibo to detect daily hotspots and analyze sentiment distributions related to these topics. Clusters of topics that describe the same issue are formed and ranked based on popularity to exploit the resulting hotspots. In [28], the authors use a clustering method to obtain candidate topics on BBS and the evolution theory to calculate the heat of candidate topics and obtain hotspots based on it. Hao and Hu [29] propose a method based on a baseline model to solve the topic drift problem of network BBS. Liu and Li [30] adopt text mining approaches based on a vector space model and k-means clustering to group Internet public opinion hotspots. Li [31] uses an emotion analysis technology to analyze the emotional polarity of network BBS Chinese texts and a k-means algorithm and the SVM to cluster the contents of posts considering each class as a hot topic. Chen et al. [32] design a similarity analysis algorithm of Internet public opinion based on information entropy, which can cluster and identify hotspots and crisis events.

The above studies provide a useful basis for this study, but there are some gaps that need to be filled, especially regarding the public opinion analysis of online posts. The unresolved issues are related to the validity and usefulness of data and the popularity of posters when used in the clustering of hotspots. Based on the above research, a data filtering strategy is introduced to improve the quality of data. Improved co-word analysis and Bisecting k-means clustering algorithms are designed using time spans and popularity to obtain more accurate results.

3. Mathematical Model of Online Post

In this section, we first build a mathematical model of online posts based on their characteristics and then use it to study hotspots.

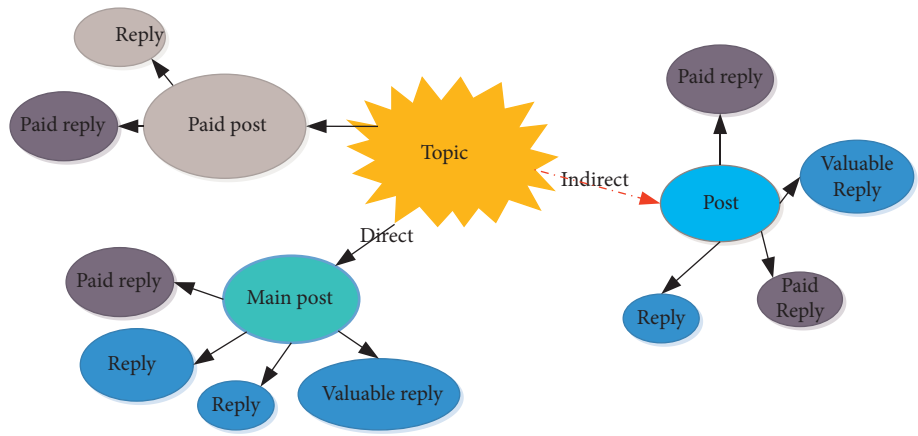


Figure 1: The structure of BBS post.

Let S_{all} be the set of all posts at time t. Let S_{valid} and S_{invalid} be the set of valid and invalid posts during period t. So, S_{all} can be represented as follows:

$$S_{\text{all}} = S_{\text{valid}} + S_{\text{invalid}}. \tag{1}$$

Assume that there are m valid posts, so S_{valid} can be expressed as $S_{\text{valid}} = \{s_1^{\text{valid}}, s_2^{\text{valid}}, \ldots, s_m^{\text{valid}}\}$... For any posts s_i, the post usually contains the first replies set $F(s_i) = \{f_1^{s_i}, f_2^{s_i}, \ldots\}$. Each $f_j^{s_i}$ has its replying content text$(f_j^{s_i})$, replying time time$(f_j^{s_i})$, and the second reply number num$(f_j^{s_i})$. The details are shown in Figure 2.

For time period t, the all topic set is

$$\begin{aligned} T_{\text{all}} &= T + T_{\text{similar}}, \\ T &= \{T_1, T_2, \ldots, T_n\}, \end{aligned} \tag{2}$$

where n is the size of the valid topics set T and T_{similar} is similar to T on the topic set in period t. In other words, T_{similar} and T have similar keywords.

It is known that all posts (S_{all}) during a period have multiple topics; therefore, getting the post relevant to the one topic is the basement for extracting the hotspots. Each topic $T_i \in T$ corresponds to the keywords set$(X_{\text{all}}(T_i))$, which contains similar keywords' set $(X_{\text{all}}(T_i) = X(T_i) + X_{\text{similar}}(T_i))$. The valid keywords set has n_i valid keywords, and we can assume that its keyword set is $X(T_i) = \{x_1(T_i), x_2(T_i), \ldots, x_{n_i}(T_i)\}$ $((x_i(T_i) \in X(T_i), 1 \le i \le n))$.

We use the notation sT_i to indicate that post $s\,(s \in S_{\text{all}})$ is relevant to T_i. That is, s contains the keywords of T_i. We can use the following equation to define this relationship between a post and a topic:

$$f(s, T_i) = \begin{cases} 1, & sT_i, \\ 0, & \text{otherwise}. \end{cases} \tag{3}$$

According to (3), the relevant posts are the post contents that contain the keywords of the one topic. Therefore, we can obtain the relevant posts set $S_{\text{all}}(T_i)$ of topic T_i as

$$S_{\text{all}}(T_i) = \{s_{\text{all}}(T_i) | f(s, T_i) = 1\}. \tag{4}$$

Definition 1. *Post lifespan* $t(s)$ is

$$t(s) = \text{day}\left(\text{time}_{f_{\text{end}}} - \text{time}_{f_0}\right), \tag{5}$$

where time_{f_0} and $\text{time}_{f_{\text{end}}}$ represent the time of the first and last reply of post s during a certain time, respectively, and day$(\cdot)$ is the number of days disregarding hours and minutes.

Definition 2. *The reply number* (the total number of replies of one post) is equal to

$$\text{TR}(s) = \sum_{j=1}^{|F|} \text{num}\left(f_j^s\right) + |F|, \tag{6}$$

where TR$(s)$is the total number of replies of post s and $|F|$is the total number of the first replies.

Definition 3. *Post participants* (p) are the post creator and repliers. Let $P(s_i)$ be the participant's participation number:

$$P(s_i) = P_{\text{first}}(s_i) + P_{\text{second}}(s_i), \tag{7}$$

where $P_{\text{first}}(s_i)$ and $P_{\text{second}}(s_i)$ are the first repliers and second repliers, respectively.

Definition 4. (degree of participation (DoP)). The degree of participation of post s_i is the ratio of the reply number and participant reply number

$$\delta_i = \frac{P(s_i)}{\text{TR}(s_i)}. \tag{8}$$

Definition 5. (minimum number of replies (MNR)). The MNR is the least reply limitation of the post s_i. Let ε_i be the MNR of the post s_i. MNR aims to simplify the valid post dataset without considering the posts with no replies and few replies.

It is clear that different posters and repliers have different influences. To calculate such influence, we use the popularity of a participant as the value. The popularity of a participant mostly depends on two factors: the frequency and the

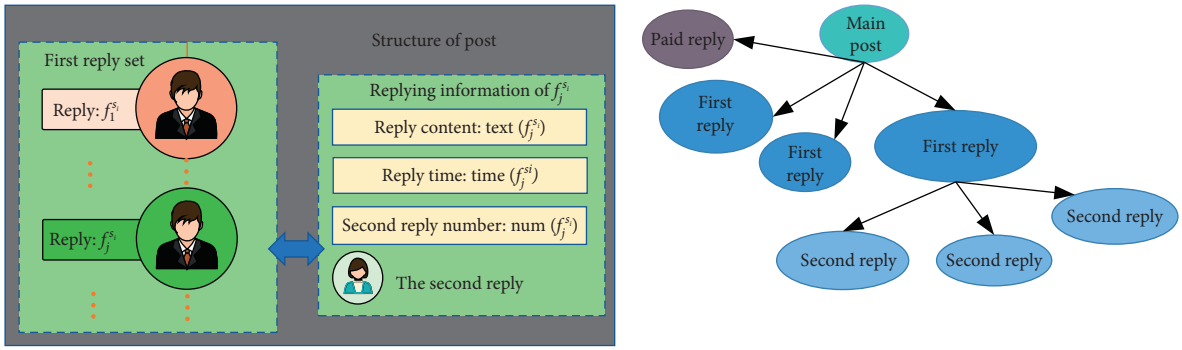


Figure 2: The framework of a post or BBS.

popularity of replies. The frequency can be expressed as the average number of participants in posts, and the popularity can be expressed as the number of replies per day or per post. Accordingly, the post participant p's popularity (forces(p)) can be expressed as

$$\text{forces}(p) = a \cdot fr + b \cdot \frac{1}{N} \cdot TSP, \tag{9}$$

where fr is the frequency of the posts of participant p, N is the total number of the discussion posts of p, TSP is the total number of repliers to all discussion posts of p, and a and b are the coefficients corresponding to the frequency and the total number of repliers, respectively.

We use different values to denote different post values for a topic. The value of a post is mainly determined by two criteria: the average number of replies and the popularity of the participant. Accordingly, the following formula is used to calculate the value:

$$weight(s) = \alpha \cdot TR(s) + \beta \cdot \sum forces(p(s)), \tag{10}$$

where α and β are the coefficients corresponding to the average number of replies and popularity, respectively.

Based on the above discussion, we can calculate the topic hot degree (HD)hot(T)as

$$\text{hot}(T) = \vartheta \cdot |S(T)| + \theta \cdot \sum_{i=1}^{m} \text{weight}(s_i), \tag{11}$$

where $S(T)$ is the valid post set about topic T and ϑ, θ are the coefficients of the total number of posts and post values, respectively.

Definition 6. Hotspot is the topic that gets the maximum value of the hot degree. For the topic set $T = \{T_1, T_2, \ldots\}$, the problem of hotspot search becomes

$$\begin{aligned} &\arg\max(\text{hot}(T_i)), \\ &\text{Subject To} : s_i \in S(T_i), \\ &T_i \in T, \\ &X(T_i) \in X_{\text{all}}(T_i), \end{aligned} \tag{12}$$

where $S(T_i)$ is the valid post set of topic T_i and the constraint conditions in equation (12) are restricted vales scope of the post, topic, and keywords, respectively.

Definition 7. Hot post is the post with the maximum value of a post. And, the hot post $s_{\max}$of topic T_i can be expressed as follows:

$$\begin{aligned} s_{\max}(T_i) = \arg\max(\text{weight}(s_i)), \\ \text{Subject To} : s_i \in S(T_i). \end{aligned} \tag{13}$$

It is clear that the main problem related to formulas (12) and (13) is to obtain valid posts, replies, and keywords.

4. Spam Data Filtering Mechanism and Improved Cluster Hotspot Monitoring Algorithm

To determine the hotspots of BBS, we use a filtering mechanism to obtain more valuable data and an improved cluster hotspot monitoring algorithm to find hotspots. We focus on text data filtering, extracting keywords, constructing the common word matrix, and searching the hotspots and hotspots.

The main process involves the following steps: the identification of postspamming and fake replies to increase the post and reply values, the application of a text rank-based keyword extraction algorithm to calculate the PageRank Value of the candidate keywords and obtain their PR values, the determination of the keywords based on posts' PR (PageRank) values, and the construction of the co-word matrix for these keywords, as shown in Figure 3. As a result, we determine the hotspots by sorting the above results.

4.1. Post Filtering. Paid posts and invalid replies are well-known phenomena in BBS networks. However, they affect the results of hotspot and hot postsearch. Accordingly, we must filter out invalid data. We adopt the following rules.

> Rule 1(degree of participation): if the degree of participation (DoP) is above a predefined constant, we consider the post as a paid post or spam post. And, we delete the post from the post set. The DoP is the ratio of

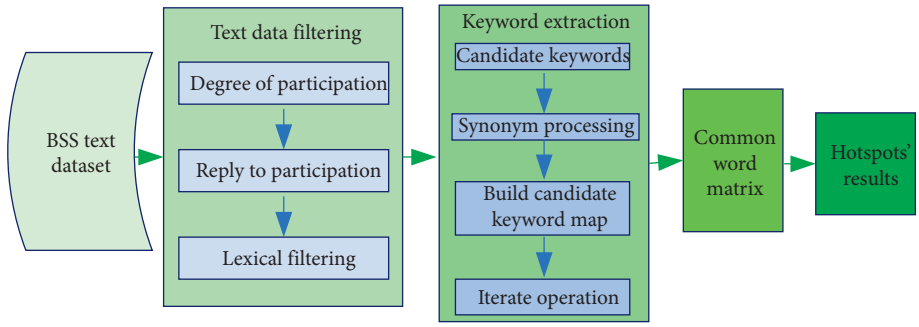


Figure 3: The main steps of the proposal strategy.

the participant's reply number and the total reply number of the post. Rule 1 can be formulated as follows:

$$S = \begin{cases} S_{\text{all}} \backslash \{s_i\}, \text{ if } \delta_i > \delta, \\ S_{\text{all}}, \qquad \text{otherwise}, \end{cases} \tag{14}$$

where δ is the DoP constant.

Rule 2 (MNR): if the MMR of the post is less than a predefined constant, we consider the post as an invalid post or spam post. And, we delete the post from the post set. The MNR is the ratio of the total reply number and the participant's number of the post. Rule 2 can be formulated as follows:

$$S = \begin{cases} S_{\text{all}} \backslash \{s_i\}, \text{ if } \varepsilon_i < \varepsilon, \\ S_{\text{all}}, \qquad \text{otherwise}, \end{cases} \tag{15}$$

whereε is the MNR constant.

We use Rule 1 and 2 to filter out paid or spam posts. It is known that, in a real BBS, there are many fake replies, which are not related to the topic, such as advertising. Such replies must be deleted from the post set as well.

Rule 3(lexical filtering): the predefined vocabulary set is denoted as $A = \{a_1, a_2, \ldots\}$for topic T_i. If a reply text ($\text{Text}_{\text{reply}}(f_{ij})$) does not contain an element of$A$, it is considered invalid and gets deleted from the replies set. The rule for getting a valid reply can be described as follows:

$$F = \left\{ S = \begin{cases} F_{\text{all}} \backslash \{f_{ij}\}, & \text{if } \text{Text}_{\text{reply}}(f_{ij}) \triangleright A, \\ F_{\text{all}}, & \text{otherwise}, \end{cases} \right. \tag{16}$$

where $\text{Text}_{\text{reply}}(f_{ij}) \triangleright A$ means the reply text contains a predefined vocabulary element of A, F_{all} is all reply for the topic T_i, and f_{ij} is the jth reply text of topic T_i.

Algorithm 1 shows the details of the post-text filtering for the topic T_i. In algorithm 1, the text data filtering can be mainly divided into the following steps. Firstly, the relevant post set of T_i can be obtained with equations (3) and (4). Then, according to Rule 1 and 2, the degree of participation and the minimum number replies are employed for deleting the invalid posts which are beyond the constraints of (14) and (15). Moreover, using Rule 3, the invalid replies are selected and removed from the reply set F_{all}. Finally, the valid post and replying sets (S, F) are returned.

4.2. TextRank-Based Keyword Extraction Algorithm. Based on Section 4.1, we can obtain valid posts and replies by adopting the filtering mechanism. In this section, we further extract keywords based on their text ranks. The main steps are as follows.

Step 1: we divide the text of a reply into a word list. Then, we order the words in the list. Namely, $\text{list}(V) = [v1, v2, \ldots]$.

Step 2: after filtering the element of list(V) according to the following rules, we obtain the list of candidate keywords.

Step 3: we use the following synonym processing rule to build candidate keywords.

Rule 4 (synonym processing): let C be the synonym keywords set:

$$C = \{C_1[c_1(\text{main}), C_1(\text{syn})], C_2[c_2(\text{main}), C_2(\text{syn})], \ldots\} \tag{17}$$

where $c_i(\text{main})$ and $C_i(\text{syn})$are the main word and its synonym set, respectively. If a word is a synonym, we use the main keyword to replace it and then merge the same words and build candidate keywords (list(X)).

Step 4 (build candidate keyword map): the candidate keyword map G=(i, list(i)), where i is the candidate keyword, list (i) is the set of words co-existing with i in the window, and, for the word j in list (i), the co-occurrence number between i and j is denoted as weight w_{ij}.

Step 5 (iterate operation): we set the number of iterations (L), according to the page rank algorithm [33, 34], we can calculate the page rank value (PR value) of each word and then construct a sequence according to the reverse order of PR values. The formula for the PR value is

Input: S_{all}, ε, A,δ
Output: $S(T_i)$, $F(T_i)$
Initialization $S = S(T_i) \leftarrow S_{\text{all}}(T_i)$, $F = F(T_i) \leftarrow F_{\text{all}}(T_i)$, $S_{\text{all}}(T_i) \leftarrow S_{\text{all}}$
Computing the relevant post set $S_{\text{all}}(T_i)$of T_i//According to equations (3) and (4)
  for $i = 1$: $|S|$//Degree of participation filtering
    Calculate $P(s_i)$
    $\delta_i \leftarrow P(s_i)/\text{TR}(s_i)$
    if $\delta_i > \delta$
      $S \leftarrow S/s_i$
    end if
  end for
  for $i = 1$: $|S|$//MNR filtering
    Calculate $\text{TR}(s_i)$
    $\varepsilon_i \leftarrow \text{TR}(s_i)$
    if $\varepsilon_i < \varepsilon$
      $S \leftarrow S/s_i$
    end if
  end for
for $i = 1$: $|S|$
  for $j = 1$: $|F|$//Lexical filtering
    if $\text{Text}_{\text{reply}}(f_{ij}) \triangleright A$
      $F \leftarrow F/f_i$
    end if
  end for
end for
Return $S(T_i)$, $F(T_i)$

Algorithm 1: The post-text filtering.

$$\text{PR}(i) = (1 - d) + d^* \sum_{j \in \text{list}(i)} \frac{\omega_{ij}}{\sum_{l \in \text{list}(j)} \omega_{jl}} \text{PR}(j), \quad (18)$$

where PR(i) refers to the PR value of keyword i, j denotes the keywords co-existing with i, l denotes the keywords co-existing with j, and d is the damping coefficient.

4.3. Common Word Matrix for Obtaining the Same Hotspots and Hot Posts. Common word matrix: n keywords are selected according to their PR values (Section 4.2). The keyword set is $W = \{w_1, w_2, \ldots, w_n\}$. The position of the co-word matrix corresponds to the semantic distance between two keywords. The formula for the semantic distance $\text{dist}(w_i, w_j)$ between two keywords (w_i, w_j) is

$$\text{dist}\left(w_i, w_j\right) = \frac{1}{\text{count}\left(w_i, w_j\right) + 1}, \quad (19)$$

where $\text{count}(w_i, w_j)$ represents the number of co-occurrence events between keywords (w_i, w_j). The smaller the semantic distance between two keywords is, the more likely the two keywords belong to the same hotspot. Therefore, the common word matrix (CA) can be represented as

$$CA = \begin{pmatrix} \text{dist}_{11} & \cdots & \text{dist}_{1n} \\ \vdots & \ddots & \vdots \\ \text{dist}_{n1} & \cdots & \text{dist}_{nn} \end{pmatrix}. \quad (20)$$

Searching the hotspots and hot posts: the common word matrix CA is transformed into a point set. Then, all the points are treated as the first cluster, and the first cluster is divided into two parts. Select each cluster that can minimize the SSE (sum of squared errors) value and divide it into two new clusters. This loop continues until the number of clusters equals the predefined number K.

The hotspots and keywords are obtained based on the above steps. Then, we can obtain the hot post using equation (10). By using the strategy explaining in equation (11), we can get the topic hot degree for every topic. Also, the hotspot and hot post by sorting can be identified.

5. Experimental Results

Experiments were conducted to evaluate the performance of the proposed algorithm using a real dataset. The results of the experiment are used to analyze the proposed approach. This section covers the simulation parameters, setup, and results.

5.1. Dataset and Experiment Setup. Dataset: the dataset is gathered from three online post websites (W1 (Baidu Tieba post): https://tieba.baidu.com; W2: https://bbs.tianya.cn; W3: http://www.xici.net). The three websites are the most famous online BBS platforms in China, which have more than 150 million active users in 2020. The post textual data was collected from these BBS and covers the whole year of 2018. Figure 4 shows a screenshot of the online post of W1, which is a classical online community post based on textual data.

To test the proposed strategies, we select three typical subjects ("Computer game" (S1), "Exam" (S2), and

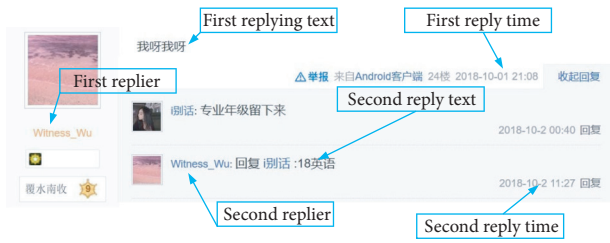


Figure 4: The screenshot of a Baidu Tieba post.

"Nanfang College" (S3)) from the above BBS websites. The data was obtained using a crawler. Meanwhile, data visualization software was designed for analyzing these textual data of BBS, and the data filtering algorithm was used in the software, as shown in Figure 5. The dataset obtained has 16,373 posts and 100,197 replies from January 1, 2018, to December 31, 2018. The parameters of the experiment are shown in Table 1.

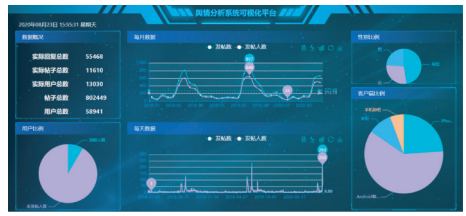


Figure 5: The data visualization software screenshot of BBS.

5.2. Results' Analysis. Valid posts and replies: the proposed data filtering mechanism is used to obtain valid data from the dataset. Figure 6 shows the results of the valid posts and replies of the above three subjects (S1, S2, S3). It is a well-known fact that, by using filtering strategies, we can effectively delete spam posts and replies. Figure 6(a) provides the comparison of the results of our filtering rules and the raw data in some different subjects for different subject posts. Our mechanism can decrease the number of posts by 109 and 513 by using Rules 1 and 2, respectively, compared to the raw data that was not filtered in S1. Accordingly, by using our filtering rules, more than 13% of the invalid post is obtained. Figure 6(b) shows the results of the filtering of online post repliers based on the proposed methods in different subjects by using rule 3. Similarly, the methods can effectively filter out invalid replies. Particularly, Rule 3 can filter more than 30% invalid replies.

The results demonstrate that the proposed filtering mechanisms can decrease the number of invalid posts and replies. Also, the filter can reduce datasets and improve the efficiency of searching for hotspots. Furthermore, the results show that the proposed data filtering algorithm has different post and replies effects on a different subject. In other words, the larger the scope of the subject, the bigger the post and replies. Topics with a wide scope of topics are more likely to have spam posts and replies.

Precision: for verifying the data filtering algorithm performance of precision, the part raw data (10%) of the subject of S3 is selected. Then, these BBS data are filtered by the manual and the proposed data filtering algorithm, respectively. Figure 7 shows the precision percentage results of posts and replies of subject S3 by using the proposed method in different BBS websites. From the results, it is easy to get that the precision of filtering posts is more than 92%, and the precision of filtering replies is large than 85%. The results demonstrate that the proposed data filtering algorithm has a good effect on the precision of spam posts and replies.

Computing time: computing time is an important metric to evaluate the performance of the data filtering algorithm. Therefore, the computing time results to collect the number of users in different subjects is given in raw and filtered data, as shown in Figure 8. It is easy to get that the S1 spends the most computing time in three subjects. And, the proposed filtering can decrease more than 15% computing time. In other words, the filtered data used less time to search the number of users than the raw dataset in all subjects. The results show that the presented strategy can save more computing time by using a data filtering algorithm.

Hot degrees: we use hot degrees to search for hotspots and posts. When the hot degree of a post reaches 3, we consider it a hotspot. After calculating hot degrees and searching for hotspots, five hotspots were selected based on their hot degrees. Hot degrees of different topics can be obtained using the hot degree calculation method. Meanwhile, the same five hotspots of the maximum number of post and repliers are calculated in the same dataset. The results of different metrics are shown in Figure 9. Figure 9(a) provides the hot degrees of different topics from the 90th to 105th days. The values of the hot degrees of the five topics are 3.6, 5.6, 5.6, 11.75, and 5.08. It is noticeable that topic 4 is the hottest topic during this period. Figures 9(b) and 9(c) show the total numbers of post and repliers in the same time period. It can be seen that topic T1 is the hotspot using the different degree, and it has the highest values of the three metrics. Namely, the hot degree can reflect the hotspots of online posts.

The values of hot degrees on subject S1are obtained in different datasets from different websites. Figure 10 shows the hot degree results from the 5th to 235th days related to topic 1. From Figure 7, it can be seen that topic 1 has three

Table 1: The experiment parameters.

Parameters	Values	Descriptions
A	0.7	The coefficients of the frequency of repliers
b	0.3	The coefficients of the total number of repliers
α	0.9	The coefficients of the average number of replies
β	0.1	The coefficients of the popularity of replier
ϑ	0.1	The coefficients of the total number of posts
θ	0.9	The coefficients of the post value
δ	0.5	The DoP constant value
window	5	The number of accommodated keywords
L	100	The number of iterations
d	0.85	The damping coefficient
K	3	The number of cluster centers
ε	5	The minimum number of replies constant value

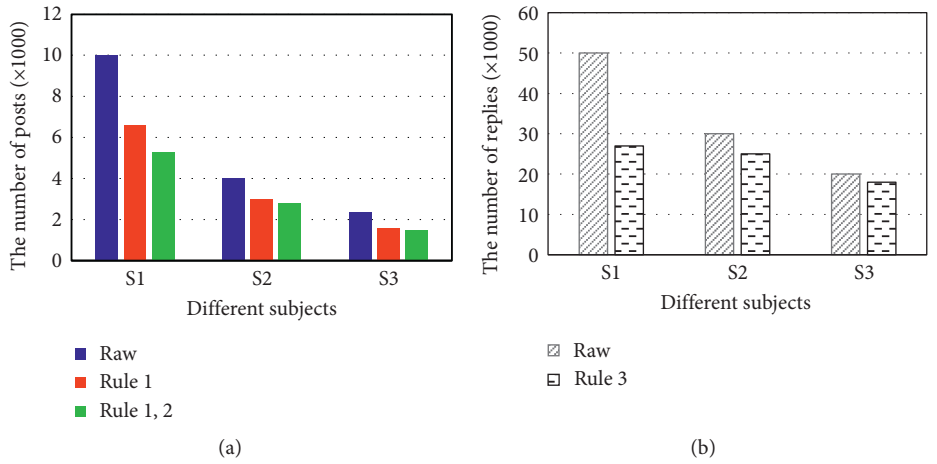


Figure 6: The comparison of valid posts and replies. (a) Filtering of posts. (b) Filtering of replies.

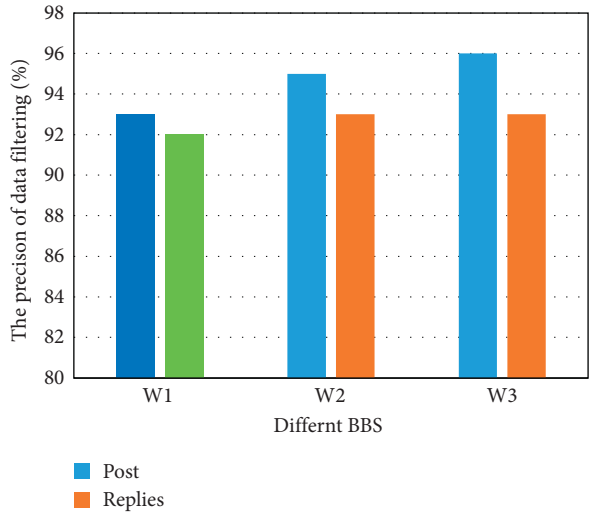


Figure 7: The comparison of precision of data filtering.

peaks at the 60th, 120th, and 190th days, and the hot degree of three websites has the same trend in different online post websites during the monitored period. The proposed hot degree can directly reflect fluctuation trends. Specifically, a hot degree can demonstrate the trend in terms of repliers and users, as in our strategies we merge post users and replies. In summary, the proposed method can effectively solve the social media hotspot problem.

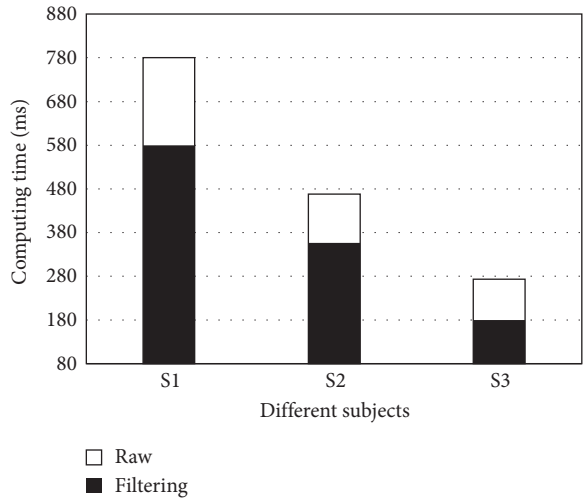


Figure 8: The results of computing time.

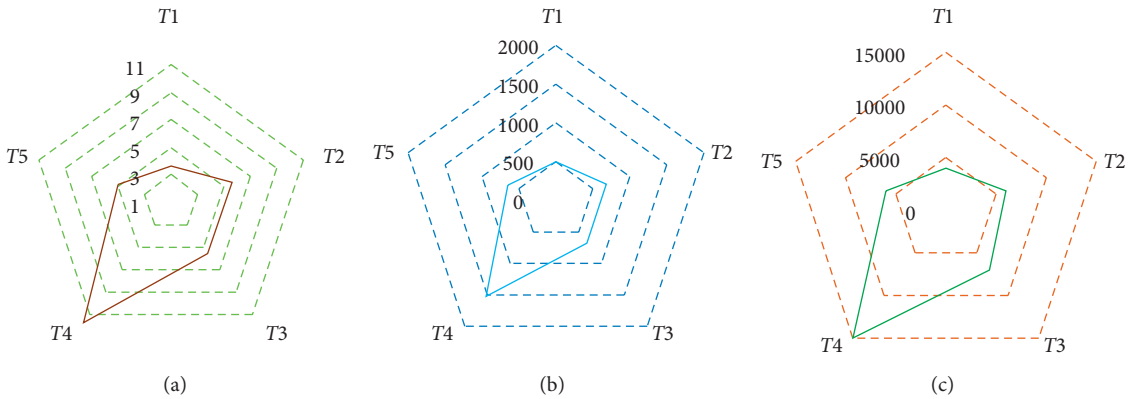


Figure 9: The results on the hot degrees (a), the number of post (b), and repliers (c) for five topics.

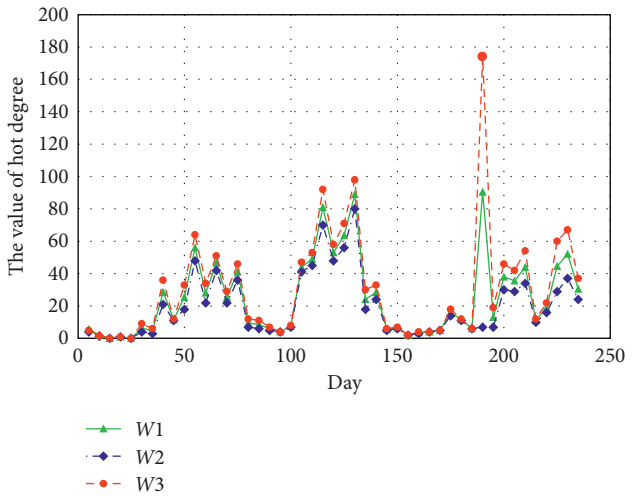


Figure 10: The hot degree results in different datasets.

6. Conclusion

The online posts have become public platforms for expressing personal opinion, so their monitoring and online hot topic search gained more significance. Considering the weight of different users, the extraction of hotspots from massive textual data with spam data become one of the important bases for study the public opinion of the social network. By collecting and analyzing text information on online posts, current hotspots can be obtained. This article adopts a data filtering mechanism, common words, and clustering technology for online hotspots search, using a time span, poster popularity, and PR values. Then, hot degree is used to evaluate the hotspots of online posts based on the number of replies and the popularity of the participant. The proposed methods are implemented and applied to a BBS dataset. The results show that the proposed method can effectively filter out invalid data, compress datasets, save more computing time, and improve performance. At the same time, the results demonstrate that the proposed method and hot degree can also reflect changes in the trend of the hotspots of online posts.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Hui-Ru Cao conceived and wrote the manuscript; Songyao Lian analyzed the data and performed the experiments; Chou-Jun Zhan and Xiaomin Li analyzed the experimental results.

Acknowledgments

This work was supported by the Ministry of Education in China Liberal Arts and Social Sciences Foundation under Grant no. 20YJCZH004, Natural Science Foundation of Guangdong Province of China under Grant 2019A1515011346, Featured Innovation Projects of Guangdong Province universities of China under Grant no. 2019GKTSCX075, and National Science Foundation of China Project under Grant no. 61703355. This work was also partially supported by colleagues at the Department of Electronic Communication and Software Engineering of Sun Yat-sen University.

References

[1] W. Hanson, "A global internet: the next four billion users," *New Space*, vol. 3, no. 3, pp. 204–207, 2015.

[2] A. Zubiaga, A. Aker, K. Bontcheva et al., "Detection and resolution of rumours in social media: a survey," *Acm Computing Surveys*, vol. 51, no. 2, 2018.

[3] J. Katz and P. Aspden, "Motivations for and barriers to Internet usage: results of a national public opinion survey," *Internet Research*, vol. 7, no. 3, pp. 170–188, 1997.

[4] J. Zeng, S. Zhang, C. Wu, and J. Xie, "Predictive model for internet public opinion," in *Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, pp. 1–11, IEEE, Haikou, China, 2007.

[5] D. Pokotylo, *Online Public Opinion and Archaeological Heritage Conservation: A Case Study from Western Canada*, pp. 35–48, Relevance and Application of Heritage in Contemporary Society, London, UK, 2018.

[6] X. Chen, M. Xia, J. Cheng et al., "Trend prediction of internet public opinion based on collaborative filtering," in *Proceedings of the International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pp. 583–588, IEEE, Xian, China, 2016.

[7] Q. Zhang and X. Zhang, "Group decision method for internet public opinion emergency with linguistic preference," in *Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 439–443, IEEE, Changsha, China, 2016.

[8] Y. Song, X. Y. Dai, and J. Wang, "Not all emotions are created equal: expressive behavior of the networked public on China's social media site," *Computers in Human Behavior*, vol. 60, pp. 525–533, 2016.

[9] C. Wang, H. Liu, and X. Guan, "Response effect assessment of internet public opinion based on fuzzy comprehensive evaluation," in *Proceedings of the International Conference on Logistics, Informatics and Service Sciences*, pp. 1–6, IEEE, Changsha, China, 2016.

[10] Y. Yang, "Research and realization of internet public opinion analysis based on improved TF-IDF algorithm," in *Proceedings Of The International Symposium On Distributed Computing And Applications To Business, Engineering And Science*, pp. 80–83, IEEE Computer Society, Hong Kong, China, 2017.

[11] E. Oster, E. Gilad, and A. Feigel, "Internet comments as a barometer of public opinion," *EPL (Europhysics Letters)*, vol. 111, no. 2, p. 28005, 2015.

[12] X. Gao and L. Fu, "Methods of uncertain partial differential equation with application to internet public opinion problem," *Journal of Intelligent & Fuzzy Systems*, vol. 33, no. 1, pp. 1–11, 2017.

[13] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Systems with Applications*, vol. 77, pp. 236–246, 2017.

[14] H. Zhu, P. Liu, and X. Shan, "Analysis of internet-based public opinion in China, 2012," *Journal of Molecular Neuroscience*, vol. 49, no. 3, pp. 614–617, 2015.

[15] S. A. Salloum, C. Mhamdi, M. Al-Emran et al., "Analysis and classification of Arabic newspapers' Facebook pages using text mining techniques," *International Journal of Information Technology and Language Studies*, vol. 1, no. 2, pp. 8–17, 2017.

[16] D. Wang, B. K. Szymanski, T. Abdelzaher, H. Ji, and L. Kaplan, "The age of social sensing," *Computer*, vol. 52, no. 1, pp. 36–45, 2019.

[17] D. Ruano-Ordás, J. Fdez-Glez, F. Fdez-Riverola, and J. R. Méndez, "Using new scheduling heuristics based on resource consumption information for increasing throughput on rule-based spam filtering systems," *Software: Practice and Experience*, vol. 46, no. 8, pp. 1035–1051, 2016.

[18] D. Ruano-Ordás, J. Fdez-Glez, F. Fdez-Riverola, V. Basto Fernandes, and J. R. Méndez, "RuleSIM: a toolkit for simulating the operation and improving throughput of rule-based

spam filters," *Software: Practice and Experience*, vol. 46, no. 8, pp. 1091–1108, 2016.

[19] D. Ruano-Ordás, J. Fdez-Glez, F. Fdez-Riverola, and J. R. Méndez, "Effective scheduling strategies for boosting performance on rule-based spam filtering frameworks," *Journal of Systems and Software*, vol. 86, no. 12, pp. 3151–3161, 2013.

[20] G. Neubaum and N. C. Krämer, "Monitoring the opinion of the crowd: psychological mechanisms underlying public opinion perceptions on social media," *Media Psychology*, vol. 20, no. 3, pp. 502–531, 2017.

[21] A. Hernandez-Suarez, G. Sanchez-Perez, K. Toscano-Medina et al., "Social sentiment sensor in twitter for predicting cyber-attacks using ℓ_1 regularization," *Sensors*, vol. 18, no. 5, 2018.

[22] Y. Dong, Z. Ding, F. Chiclana, and E. Herrera-Viedma, "Dynamics of public opinions in an online and offline social network," *IEEE Transactions on Big Data*, p. 1, 2017.

[23] W. Zhang, X. Li, H. He et al., "Identifying network public opinion leaders based on Markov Logic Networks," *The Scientific World Journal*, vol. 2014, no. 5, p. 268592, 2014.

[24] L. Zhang, C. Su, Y. Jin, M. Goh, and Z. Wu, "Cross-network dissemination model of public opinion in coupled networks," *Information Sciences*, vol. 451-452, pp. 240–252, 2018.

[25] Z. Feng, "Hot news mining and public opinion guidance analysis based on sentiment computing in network social media," *Personal and Ubiquitous Computing*, vol. 23, pp. 373–381.

[26] D. Jiang, X. Luo, J. Xuan, and Z. Xu, "Sentiment computing for the news event based on the social media big data," *IEEE Access*, vol. 5, no. 99, pp. 2373–2382, 2017.

[27] Y. Zhao, B. Qin, T. Liu, and D. Tang, "Social sentiment sensor: a visualization system for topic detection and topic sentiment analysis on microblog," *Multimedia Tools and Applications*, vol. 75, no. 15, pp. 8843–8860, 2016.

[28] D. Zheng and F. Li, "Hot topic detection on bbs using aging," in *Proceedings of International Conference on Web Information Systems and Mining (WISM'09)*, Shanghai, China, November 2009.

[29] X. iulan Hao and Y. Hu, "Topic detection and tracking oriented to bbs," in *Proceedings of 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE)*, Changchun, China, 2010.

[30] H. Liu and X. Li, "Internet public opinion hotspot detection research based on k-means algorithm," in *Proceedings of Advances in Swarm Intelligence, First International Conference, ICSI 2010*, Beijing, China, June 2010.

[31] N. Li and D.D. Wu, "Using text mining and sentiment analysis for online forums hotspot detection," *Decision Support Systems*, vol. 48, no. 2, pp. 354–368, 2010.

[32] X. G. Chen, S. Duan, and L. D. Wang, "Research on clustering analysis of Internet public opinion," *Cluster Computing*, vol. 22, no. 3, 2019.

[33] Y. Gao, X. Yu, and H. Zhang, "Overlapping community detection by constrained personalized PageRank," *Expert Systems with Applications*, vol. 173, Article ID 114682, 2021.

[34] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

Hindawi
Scientific Programming
Volume 2020, Article ID 8897244, 18 pages
https://doi.org/10.1155/2020/8897244

Research Article

HPM: A Hybrid Model for User's Behavior Prediction Based on *N*-Gram Parsing and Access Logs

Sonia Setia,[1,2] **Verma Jyoti**,[3] **and Neelam Duhan**[3]

[1]*J. C. Bose University of Science and Technology, YMCA, Faridabad 121006, India*
[2]*Faculty of Computer Applications, MRIIRS, Faridabad, India*
[3]*Faculty of Computer Science, J. C. Bose University of Science and Technology, YMCA, Faridabad 121006, India*

Correspondence should be addressed to Sonia Setia; setiasonia53@gmail.com

Received 26 July 2020; Revised 6 October 2020; Accepted 20 October 2020; Published 6 November 2020

Academic Editor: David Ruano-Ordás

The continuous growth of the World Wide Web has led to the problem of long access delays. To reduce this delay, prefetching techniques have been used to predict the users' browsing behavior to fetch the web pages before the user explicitly demands that web page. To make near accurate predictions for users' search behavior is a complex task faced by researchers for many years. For this, various web mining techniques have been used. However, it is observed that either of the methods has its own set of drawbacks. In this paper, a novel approach has been proposed to make a hybrid prediction model that integrates usage mining and content mining techniques to tackle the individual challenges of both these approaches. The proposed method uses *N*-gram parsing along with the click count of the queries to capture more contextual information as an effort to improve the prediction of web pages. Evaluation of the proposed hybrid approach has been done by using AOL search logs, which shows a 26% increase in precision of prediction and a 10% increase in hit ratio on average as compared to other mining techniques.

1. Introduction

The World Wide Web (WWW) has become an important place for people to share information. The amount of information available on the web is enormous and is growing day by day. As a result, it is the need of the hour to develop new techniques to access the information very quickly and efficiently. For fast delivery of media-rich web content, latency tolerant techniques are highly needed, and several methods have been developed in the past decade in this regard. Among these techniques, the two most prevalent techniques are caching and prefetching. However, caching benefits are limited due to the lack of sufficient degrees of temporal locality in the web references of individual clients [1]. The potential for caching of the requested files is even declining over the past years [2]. On the other side, prefetching is defined as "to fetch the web pages in advance before a request for those web pages" [3]. The usefulness of prefetching the web pages depends upon how accurately the prediction for those web pages has been made. A good prediction model can find various applications, of which the most prominent ones are website restructuring and reorganization, web page recommendation, determining the most appropriate place for advertisements, web caching and prefetching, etc. In recent years, due to the wide scale of applications, the prediction process has gained more importance. To make predictions, several web mining techniques have been used in the past several years. Web mining [4] can be divided into three distinct areas:

(i) Web usage mining: it involves analyzing user access patterns collected from web servers better to predict the users' needs [5–7]

(ii) Web content mining: it involves extracting useful information from websites to serve the users' needs

(iii) Web structure mining: it is the study of the interlinked structure of web pages

Traditional prefetching systems make predictions based on the usage information present in access logs. They typically employ the data mining approaches like association rule mining on the access logs to find the frequent access patterns, match the user's navigational behavior with the antecedent of the rules, and then prefetch the consequent of the rules. However, this approach's problem is that a relevant page that might be of user's interest can be exempted from the prediction list if it is new or it was not frequently visited before; therefore, it does not appear in frequent rules.

On the other side, predictions based on content information present in web pages such as title, anchor text, etc. resolve these problems, but they have their own set of drawbacks. They lack the user's intent of the search, and web content alone is insufficient to make accurate predictions.

In this paper, instead of focusing only on the content, i.e., anchor texts associated with URLs (Uniform Resource Locator), the queries submitted by users recorded in web access logs have also been crucial for actual user's interest. Therefore, a hybrid prediction model (HPM) has been proposed, which incorporates both the history of the users' browsing behavior and the information content inherent in the users' queries. It is based on the Query-URL click-graph, a bipartite graph G between queries Q and URLs U, which are extracted from the access logs. Edges E in the diagram indicate the presence of clicks between queries and URLs. Weight $C_{q,u}$ is assigned to each edge, representing the aggregated clicks between query q and URL u. N-gram parsing of queries has also been used for better results as compared to unigrams. An N-gram [8] is an N-word sequence. An N-gram of size 1 is referred to as a unigram, 2-gram as a two-word sequence, also called bigrams, and size 3, i.e., 3-gram meaning a three-word sequence, trigram. For example, parsing the query "college savings plan," we get three unigrams ("college," "savings," "plan"), two bigrams ("college_savings," "savings_plan"), and one trigram ("college_savings_plan"). The reason to use the N-gram approach is that grams can capture more contextual information, which can help us to predict the frequency of such kinds of keywords.

The advantages of this prediction framework mainly lie in three aspects:

(i) First, query terms are used through the Query-URL click-graph to understand users' behavior more accurately rather than using noisy and ambiguous web page content

(ii) Second, it captures information from both usage logs and content knowledge, which increases the accuracy of prediction

(iii) Third, this framework further considers the N-gram parsing of queries, which also improves the prediction results

The paper has been organized as follows. Section 2 highlights the detailed literature review on prefetching. The proposed approach is presented in Section 3, which discusses the following:

(i) The architecture of the hybrid prediction model

(ii) The workflow of both phases, i.e., online phase and offline phase

(iii) Detailed pseudocode for the proposed method

Further, Section 4 discusses an example of the proposed work. Experimental evaluation and comparison of the proposed work with the existing approaches are provided in Section 5. Section 6 finally concludes this work with future enhancement.

2. Related Work

Web prediction is a classification problem to predict the next web page that a user may visit based on its browsing history. Several researchers have been trying to improve the prediction of users' browsing experience in the past decade to achieve the following research objectives:

(i) To improve the accuracy of prediction

(ii) To remove the scalability problem

(iii) To improve prediction time

This section talks about various techniques and methods used to develop web page predictions categorized under usage mining, content mining, and structure mining.

2.1. Prefetching Techniques Using Usage Mining. Markov model is a mathematical tool for statistical modeling, one of the popular methods used for prefetching. Generally, the Markov model's basic concept is to predict the next action, which depends on the results of previous actions. Several researchers have used this technique successfully in various literature studies to train and test user actions or predict their future behavior.

Deshpande and Karypis [9] and Kim et al. [10] investigated that high accuracy in the prediction of the next web page can also be achieved by using higher-order Markov models. Still, higher-order Markov models have high space complexity, whereas lower-order Markov models cannot capture the users' browsing behavior accurately. To solve this problem, Verma et al. [11] proposed a novel approach for web page prediction using the k-order Markov model, where the value of "k" has been chosen dynamically. In addition to this work, Oguducu and Ozsu [12] and Lu et al. [13] worked upon user sessions. User sessions were clustered and represented by clickstream trees for making predictions. But it raises a scalability problem. Further, Awad and Khalil [14] analyzed the Markov model and all-K^{th} Markov model to solve the web prediction problem to remove scalability problem. The proposed framework by [14] improved the prediction time without compromising prediction accuracy.

Zou et al. [15] found that more accurate prediction models are required; therefore, more complex prediction tasks must run. In this paper, the authors proposed the intentionality-related long short-term memory (Ir-LSTM) model, which is based on the time-series characteristics of browsing records. Further, Joo and Lee [16] proposed a framework for user-web interaction called WebProfiler.

Basically, it predicts the user's future access based on user interaction data collected by this profiler. The authors claimed that overall prediction performance using the proposed model had been improved by 13.7% on average.

Martinez-Sugastti et al. [17] presented a prediction model based on history-based prefetching approach. This model considers the cost of prediction in terms of cache hits and cache misses of the forecast to train the prediction model so that more accurate results can be achieved based on the previous cache hits. The authors claimed that, by using this model, the precision of prediction had been improved, and latency has been reduced. Veena and Pai [18] proposed the "Density Weighted Fuzzy C Means" clustering algorithm to cluster similar user's access patterns. This algorithm can be used for the recommendation system as well as the prefetching system.

2.2. Prefetching Techniques Using Content Mining. Keeping content at the epicenter of the research approach, Venkatesh [19] proposed a prefetching technique that used hyperlinks and associated anchor texts present in the web page for predictions. The probability of each link was computed by applying Naïve Bayes classifier on the anchor text concerning keywords of the user's interest. The connections with higher chances were chosen for prefetching. Further, Setia et al. [20] extended this work by considering the semantic preferences of the keywords present in the anchor text associated with the hyperlinks.

Researchers [21–23] proposed a semantically enhanced method for a more accurate prediction that integrated the website's domain knowledge and web usage data.

Authors [24, 25] found that only the user's access patterns are insufficient to predict the user's behavior. The authors [24] worked upon an individual user's behavior. Authors [25] analyzed that web pages' content should also be taken into account to capture the user's interest.

2.3. Prefetching Techniques Using Structure Mining. Web link analysis [26] proved to be an important factor in performing a good quality web search. It can also calculate how the web pages are related to each other. Link analysis approaches are divided into two types: "explicit link analysis" and "implicit link analysis." Hyperlinks present on the web page are called explicit links. It has been proved by Davison [27] that hyperlink information can help a lot in web search. Web designers design the structure of the links and embed the links in the website. Therefore, in the case of the "explicit link analysis" technique, the user follows the design that the website designer was responsible for making any web page important, e.g., Kleinberg's HITS [28]. However, in the "implicit link analysis" technique, the importance of a web page is not determined by the web page designer, but it is done by the users who are accessing that web page. The higher the number of users accessing the web page, the more influential the page is. Whenever a user accesses a web page, an implicit link is developed between the user and the corresponding web page. Further, pages are visited by the user in a sequential manner, forming implicit associations one after another. So, in the latter case, the web page is essential from the user's point of view. An example of the implicit link analysis approach is DirectHit [29]. Researchers [26] used both techniques, i.e., "explicit link analysis" and "implicit link analysis," and further improved the search accuracy by 11.8% and 25.3%, respectively.

Authors [30–32] found that the poor structure of the website may degrade the performance of any algorithm which works upon the structure of the website for user navigation. Sheshasaayee and Vidyapriya [30] proposed a framework to reorganize the website using splay trees, a self-balancing data structure. Further, Thulase and Raju [32] extended this approach by using concept-based clustering. Vadeyar and Yogish [31] developed farthest first clustering-based technique to reorganize the website.

Table 1 describes in brief different methods for prefetching technique with appropriate justification in the context of research work.

A critical look at the above table highlights the fact that each of the existing prefetching techniques proposed by researchers has its drawbacks. Either these techniques are lacking in making the right set of prediction or the choice of parameters is not sufficient or the cost involved in making such predictions is very high.

2.4. Problem Statement. A precarious look at the literature highlights the following areas of improvements:

(i) Most of the techniques utilize the browsing history of users stored in client logs, proxy logs, or server logs in the literature. The information found in any type of access logs varies according to the format of the records. Administrators select the log data in their way. But due to insufficient information present in logs, inaccurate predictions are derived, rendering the prefetching approaches to work inefficiently. These techniques cannot predict those web pages which are newly created or never visited before.

(ii) Web pages' content information has also been widely used for predictions as a solution to the above-said problem. These techniques use the content information such as titles, anchor text, etc. which do not provide sufficient details of the user's interest and thus cannot be considered alone for prediction algorithms to work.

(iii) Structure mining-based prediction techniques depend only upon how website structure has been designed. The reorganization of the website structure for user navigation increases computational cost.

It leads to the following main problems of prediction:

(i) Less accurate prediction results and, therefore, less precision

(ii) Low hit ratio of predicted pages and, therefore, more consumption of network bandwidth

Table 1: Prefetching technique with various methods and their justification.

Sr. No.	Method used	Literature reference	Description	Justification in the context of research work
1.	Markov model	[9–13]	It is a well-known approach for pattern recognition. It determines the next state from the current state based on the orders of the Markov chain	The main problem is lack of prediction accuracy with lower-order chain, while high complexity with the higher-order chain. However, this approach does not suit the current research context
2.	Prediction by partial match	[15, 16, 33]	The PPM model uses a set of previous objects to predict the next item in a particular stream	It is a restricted version of Markov chain that provides prediction based on the only selected set of objects and selection of a right set of objects is a very challenging task, so this kind of vision is not also; it limits the result as it does not cover all the objects, thereby ruling it out of the scope of current work
3.	Cost function	[14, 17]	Prediction of future requests has been made based upon certain factors like the popularity and lifetime of web objects	A very less popular approach for pattern determination as the cost functions vary from time to time, thereby reducing the contribution in making the right set of prediction. So this approach is also not suitable in the context of the proposed research
4.	Data mining	[18]	It is also one of the most popular approaches in the modern era for pattern recognition of structured objects	The data mining approach consists of many techniques which are ideal for pattern generation task. But the proposed research is not working upon pattern generation task
5.	Keyword based	[19, 20, 24, 25]	Prediction is made by retrieving confidential information present in the contents of web documents	To work upon only this category is not much beneficial since it does not deal with multiple user transactions
6.	Integration of domain knowledge	[21–23]	It works by the integration of domain knowledge with other methods of prefetching; semantics are taken into account	It gives useful information based on semantics but increases prediction time as well as extra overhead
7.	Implicit link analysis	[26, 29–32]	In the "implicit link analysis" technique, the importance of a web page is determined by the users who navigate the web page	It is a significantly less popular approach for pattern determination. Extra work is required to reorganize the structure of the website as per user navigation
8.	Explicit link analysis	[26–28]	In the "explicit link analysis" technique, the importance has been given to the design that has been structured by the designer who makes any web page more important or less important	It gives useful information based on hyperlink structures of the web

To improve the prediction technique, a hybrid prediction model is proposed in this work, which utilizes the best of both the information, i.e., the usage information and the content information of the web pages. The poor structure of a website may degrade the performance of such kind of techniques. Therefore, we are not considering structure mining for our proposed approach.

3. Proposed Hybrid Model

This work uses the Query-URL click-graph concept, which enables incorporating crucial contextual information in the prediction algorithm. In general, the workflow of our proposed approach (shown in Figure 1) is carried out in two phases, which is discussed as follows:

(i) Offline phase: the offline phase works at the backend and runs periodically to update the logs. Since it is a hybrid model, the input to this phase is the access logs and the content information of the web pages. The combined data from both sources is then put to use by using various intermediary steps to make a relevant prediction of users' behavior. The output of this phase is the weighted logs (WL) that contain the weighted N-grams corresponding to the respective URLs.

(ii) Online phase: the online phase involves both the proxy and the client. While users interact with the system, the system predicts users' behavior according to the user's information. This information is matched with the information collected from the logs in the offline phase.

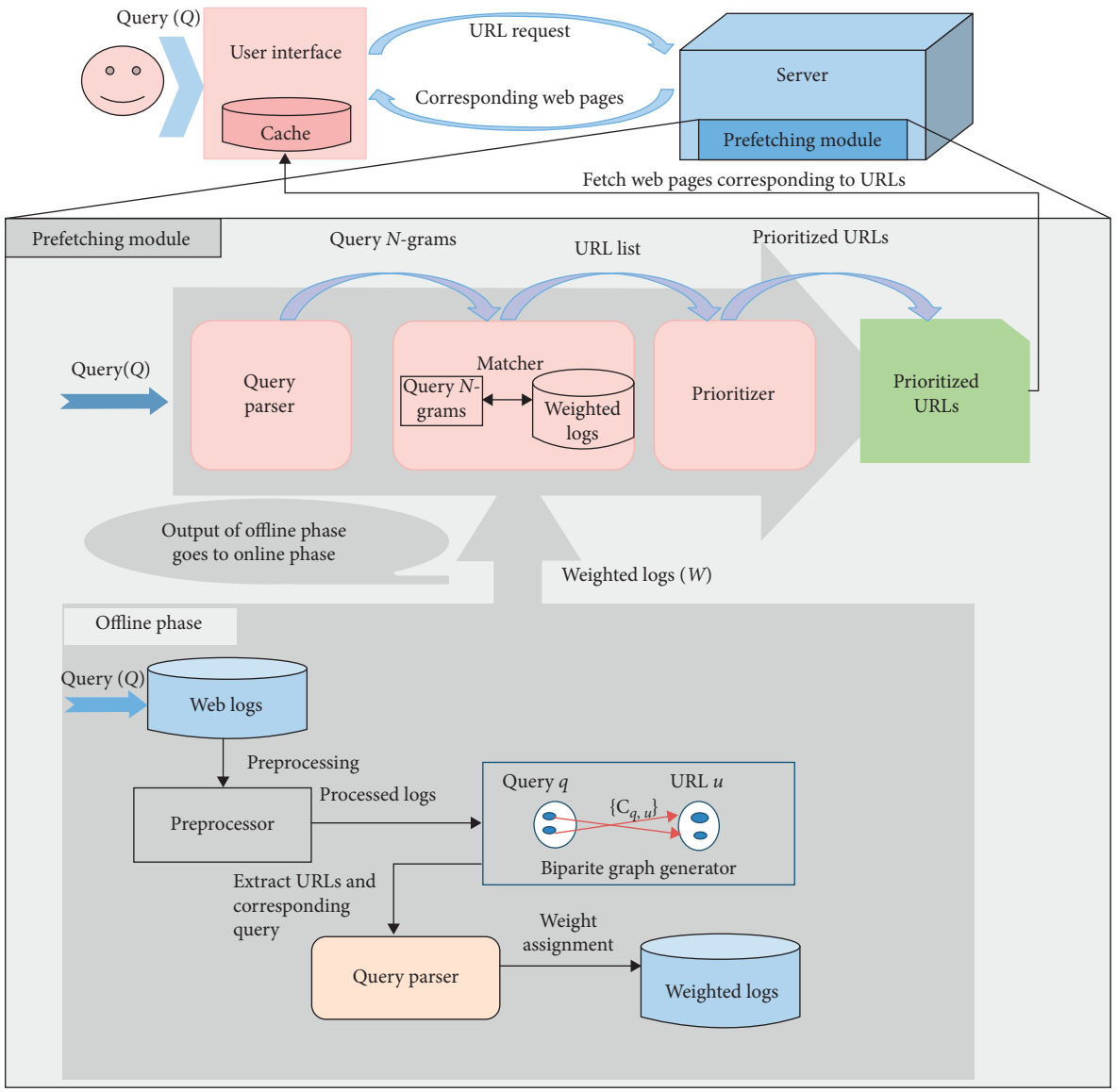


Figure 1: Architecture of hybrid prediction model.

3.1. Work Flow of the Offline Phase. This phase works in several steps, as follows:

(1) Preprocessing: initially, the offline phase considers access logs. Logs contain an entry for each request of the web pages made by the client. Various fields [34] of the records are anonymous user id, requested query, date and time at which the server is accessed, item rank, and URL clicked by the user corresponding to the requested query.

Each access log entry is preprocessed to remove stop words and extract the requested query, clicked URL corresponding to the requested query.

The processed information gets stored in the form of processed logs (PL).

(2) Bipartite graph generation: a bipartite graph between queries Q, and URLs U, taken from PL, is generated. The bipartite graph has been chosen because it helps us to improve readability. This new representation naturally bridges the semantic gap between queries and web page content and encodes rich contextual information from queries and users' click behaviors for prediction. This helps to reduce the space and computational complexity as it eliminates the need to scan the logs each time. Also, click count of the queries for the respective URLs is calculated as the graph is being generated in order to reflect the users' confidence in the query, i.e., how close the queries are connected with the clicked URLs. The edges between Q and U indicate the presence of clicks

between queries and their corresponding URLs. The generated bipartite graph is known as Query-URL click-graph (*C*-graph). The nomenclature for the generated *C*-graph is as follows:

(i) $Q = \{q_1, q_2, \ldots, q_m\}$.
(ii) $U = \{u_1, u_2, \ldots, u_n\}$.
(iii) $\langle C_{q,u} \rangle$ is an edge depicting number of clicks between Q and U.
Consider an example having $Q = \{q_1, q_2\}$ and $U = \{u_1, u_2, u_3\}$. A sample C-graph is depicted in Figure 2.
Here, the label on the Edge $\langle q_1, u_1 \rangle$, i.e., $C_{q1,u1}$, depicts that the URL u_1 has been clicked five times corresponding to the query q_1.

(3) Query parsing: queries present in *C*-graph are parsed into *N*-grams that describe the URLs' content, resulting in *N*-gram associated click-graph (NC-graph).

(4) Weight assignment: weights are assigned to each *N*-gram in the query, present in NC-graph, based on the number of times a query has been clicked, which is depicted on the edges by $C_{q,u}$ in *C*-graph. The same click count is assigned to each *N*-gram of query, i.e., $C_{n,u}$, which is equivalent to $C_{q,u}$, where $\langle C_{n,u} \rangle$ is an edge depicting the number of clicks between *N*-gram n and URL u. For example, query q_1 is parsed into *N*-grams n_1 and n_2 which results in NC-graph depicted in Figure 3. As we can see in Figure 2, $C_{q1,u1} = 5$; therefore, its *N*-grams, i.e., $C_{n1,u1} = 5$, and $C_{n2,u1} = 5$.

Corresponding to each URL "u," a weighted vector is defined that comprises the weighted *N*-gram $w_{n,u}$. Further, $W_{n,u}$ is computed by adding click count of the *N*-grams ($C_{n,u}$) coming from different queries for that URL.

Finally, weighted *N*-grams are normalized to rescale the values by using

$$W_{n,u} = \frac{w_{n,u}}{\sum_{v \in Vu} C_{v,u}}, \qquad V_u = \left\{ V \in N_q : \; N_q \in \langle q, u \rangle \right\}, \tag{1}$$

where $w_{n,u}$ is divided by the summation of click counts of all the terms corresponding to all the queries representing the URL u,where

u represents the URL

n represents one *N*-gram for the query

v is a term

V_u defines all the words belonging to *N*-grams about the different queries representing the URL u

N_q represents all the *N*-grams of the query q

$w_{n,u}$ represents weight of *N*-gram n in the URL u

$C_{v,u}$ represents click count of each term for the URL u

All the processing is done in temporary memory, and finally, it outputs weighted logs, which contain the URLs and their corresponding *N*-grams and their associated weights. The schema of access logs (AL), processed logs (PL), and weighted logs (WL) is shown in Figure 4.

The description of different attributes is given in Table 2.

It is important to note here that the offline phase runs periodically to update access logs. On every periodic update, only the fragment containing new entries in access logs is considered for further processing, and accordingly, weighted logs are updated. This job is done by the Incremental Module, a submodule of the prefetching module, as depicted in Figure 5.

3.2. Work Flow of the Online Phase. The online phase can be discussed in five major steps, as follows:

(1) Query initiation at interface: user enters a query according to his interest, which goes to the server through a proxy using the HTTP GET method. The server responds with the list of URLs corresponding to the respective query.

(2) Parser activation: while the user views the current page, the proxy server uses this query for further processing at the back end. This initializes the parser that parses this query into *N*-grams called query terms stored in set T. The resulting query terms are used to find the relevant URLs (from the weighted logs (WL)) corresponding to the respective query.

(3) Matcher activation: this phase takes as input the query terms from T from the online phase and weighted logs (WL) from the offline stage. The weights of URLs corresponding to the users' query are calculated by comparing the users' query terms T with the weighted *N*-grams of URLs in WL. This process is carried with the help of (2):

$$W_u = \sum_{t \in T} W_{t,u} * I_{t,u}, \tag{2}$$

where

W_u represents the weight of each URL,

W_t represents the weight of each term present in the URL,

$I_{t,u}$ is a vector for each URL, i.e.,

$$I_{t,u} = \begin{cases} 1, & \text{if } t \text{ present in URL } u, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

(4) Prediction list generation: these weights are then fed to the prediction unit. It prioritizes the URLs based on their weights generated in step 3. A prediction list of URLs corresponding to the user query based on this prioritization is generated.

(5) Prefetching: prefetcher prefetches the predicted URLs and stores them in the cache.

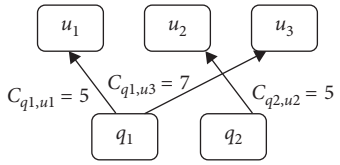


Figure 2: Example of C-graph.

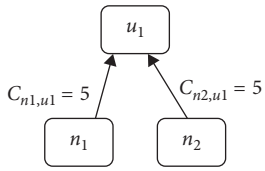


Figure 3: Example of NC-graph.

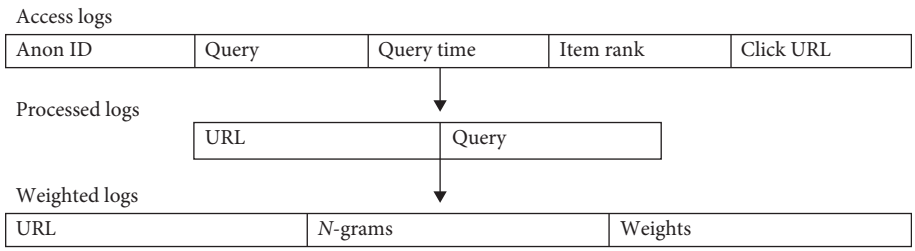


Figure 4: Schema of logs used for proposed approach.

Table 2: Attributes of schema and their description.

Attribute	Description
AnonID	An anonymous user ID number
Query	The query issued by the user
QueryTime	The time at which the query was submitted for search
ItemRank	If the user clicked on a search result, the rank of the item on which they clicked is listed
ClickURL	If the user clicked on a search result, the domain portion of the URL in the related work is listed
N-grams	Parsed query in N-grams
Weights	Count of a query clicked for URL

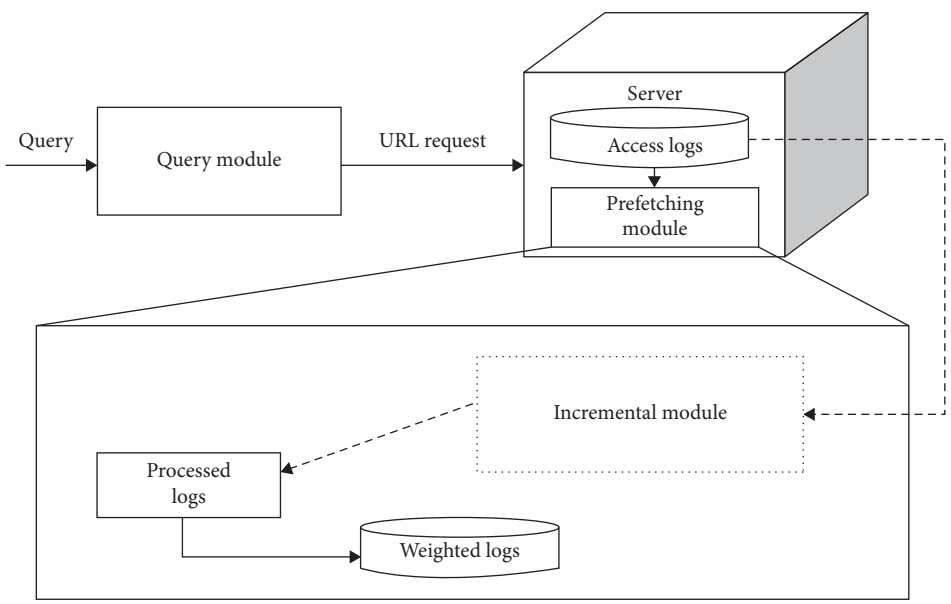


Figure 5: Incremental Module.

3.3. Pseudocode for Proposed Algorithm. The pseudocode for the proposed approach is as: Given in Algorithms 1–6.

4. Example Illustration

This section explains the offline and an online phase steps with the help of some sample of URLs, submitted queries present in the processed logs, and their respective clicks, i.e., the number of times URL has been clicked.

4.1. Preprocessing Phase

(i) In the first phase, preprocessing is done by removing stop words. A sample of preprocessed logs is shown in Table 3.

4.2. Bipartite Graph Generation Phase

(i) Calculate click count $C_{q,u}$ for each pair of query q and URL $u<q \in Q,\ u \in U>$ using processed logs. After calculating the click counts, a Query-URL click-graph (C-graph) is generated as discussed in step 5 of algorithm BipartiteGraphGen (); e.g., let $<q_1,\ u_1>$ edge is created with label $C_{q1,u1}$, i.e., 10. Similarly, $<q_5,\ u_1>$ and $<q_8,\ u_1>$ edges are created with labels 10 and 5, respectively.

(ii) Further in step 7 of BipartiteGraphGen (), the queries are parsed into N-grams by using $n = 3$ as shown in Figure 2; e.g., q_5 is parsed into 3-grams (gov, college, gov-college).

(iii) According to the algorithm's next step 8, N-gram associated click-graph (NC-graph) is generated as depicted in Figure 6.

4.3. Weight Calculation Phase

(i) The same click count is assigned to each N-gram in the query for each URL based on click count of queries as in step 6 of WeightCalculator(), e.g., with the URL u_1 associated queries, and their labels are $q_1 \longrightarrow 10,\ q_5 \longrightarrow 10,\ q_8 \longrightarrow 5$.

Against each query, parsed N-grams are
$q_1 \longrightarrow$ {ymca}, $q_5 \longrightarrow$ {gov, college, gov-college}, $q_8 \longrightarrow$ {best, college, best-college}.
Thus, each N-gram will get the respective label of its query, i.e., (ymca:10), (gov: 10, college: 10, gov-college:10), (best:5, college:5, best-college:5).

(ii) In the next step, weights are assigned to each distinct N-gram associated with URL u in NC-graph by adding click count of the N-grams coming from different queries for that URL; e.g., weighted N-grams corresponding to URL u_1 are (ymca: 10, gov: 10, college: 15, gov-college: 10, best: 5, best-college: 5)

(iii) Perform normalization as in step 10 of WeightCalculator() $W_{\text{ymca},u1} = 10/(10 + 10{+}15 + 10{+}5 + 5) = 0.22$. The normalized weighted N-grams for their respective URLs are shown in Figure 7.

4.4. Online Phase

(i) In the online phase, when the user submits a query, e.g., "ncrgov college," it is parsed in 3-grams as discussed in step 3 of Matcher () algorithm and shown in Figure 7.

(ii) Further, weights of URLs are calculated corresponding to the user's query as per step 7 of the Matcher() algorithm, e.g., $W_{u1} = 0 + 0.22 + 0.33 + 0.22 + 0 + 0 = 0.77$. To calculate the weight of u_1, weights of the user's query terms (ncr, gov, college, ncr-gov, ncr-college, gov-college, ncr-gov-college) are taken from the weighted N-grams of the URL u_1: (ymca: 0.22, gov: 0.22, college: 0.33, gov-college: 0.22, best: 0.11, best-college: 0.11) if they are present in that URL; otherwise, it is considered 0.

(iii) Based on the calculated weights of URLs, the system gives the prioritized list of URLs, as depicted in Figure 8. For further processing, the prioritized list will be passed to the prefetching engine.

Thus, the proposed approach predicts by considering the content information and the information collected using logs instead of directly deriving the frequent patterns from the access logs. Therefore, this process indicates those web pages that are not frequently visited before making more accurate predictions.

In the next section, the proposed approach's performance evaluation is carried out with a unigram approach. It has been observed that the proposed hybrid approach significantly improves performance.

5. Experimental Evaluation

The effectiveness of the proposed prediction model is illustrated by implementing and testing with a large dataset. To explore the performance of prediction, Microsoft Visual Studio 12.0 in conjunction with SQL server 2012 is used. In this section, we first list the measures for the performance evaluation of prediction and then present the impact of the n-grams followed by comparing experimental results.

5.1. Training and Testing Data. To run the experimental cases, American Online (AOL) search logs are collected for three months spanning from 01 March 2006 to 31 May 2006. This dataset consists of 20 M web queries collected from 650 k users over three months. The dataset [35] includes (AnonID, Query, QueryTime, ItemRank, ClickURL).

```
Input: access logs (AL)
Output: Weighted N-grams stored in weighted logs (WL) of order m × n
Begin
(1)  Read (AL);
(2)  PL ← Preprocess (AL);//PL = Processed Logs
(3)  NC-graph ← BipartiteGraphGen (PL); //NC-graph = N-gram associated click-graph
(4)  WL ← WeightCalculator (NC-graph); //WLis weighted logs stored in form of m × n weight matrix
(5)  Return (WL);
End
```

Algorithm 1: Weight generator.

```
Input: access logs (AL)
Output: processed logs (PL)
Begin
(1)  Read AL;
(2)  Extract session id, query, clicked URL from AL;
(3)  PL ← Remove stop words from each log record;
(4)  Return PL;
End
```

Algorithm 2: Preprocess.

```
Input: processed logs (PL)
Output: N-gram associated click-graph (NC-graph)
Begin
(1)  Read (PL);
(2)  Q ← Read queries from PL;
(3)  U ← Read URLs from PL;
(4)  Calculate click count C_{q,u} for each pair ⟨q ∈ Q, u ∈ U⟩ using PL;
(5)  C-graph ← create an edge between ⟨q, u⟩ with label C_{q,u};
(6)  For each query q ∈ Q do
(7)  N_q ← Parser (q); //parsing of query into N-grams
(8)  NC-graph ← Create an edge between ⟨q, N_q⟩
(9)  EndFor
(10) Return (NC-graph);
End
```

Algorithm 3: BipartiteGraphGen.

```
Input: query q
Output: N-grams associated with query (q), i.e., (N_q)
Begin
(1)  Read q;
(2)  N_q ← Extract N-grams from q;
(3)  Return N_q;
End
```

Algorithm 4: Parser.

Input: N-gram associated click-graph (NC-graph)
Output: weighted N-grams corresponding to distinct URLs stored in matrix WL
Begin
(1) Create a matrix WL of order $m \times n$//$m \longrightarrow$ *no. of distinct N-grams of all the queries of PL and* $n \longrightarrow$ *no. of URLs of PL*
(2) $W_{i,j} = 0$;//elements of WL
(3) For each URL $u \in U$ in NC-graph do
(4) $W_{n,u} = 0$//weight of N-gram associated with query q corresponding to URL u
(5) For each N-gram $n \in N_q$ in NC-graph do
(6) $C_{n,u} = C_{q,u}$; //$n \in N_q$
(7) $w_{n,u} += C_{n,u}$;
(8) End For
(9) For each N-gram $n \in N_q$ in NC-graph do
(10) $W_{n,u} = w_{n,u} / \sum_{v \in Vu} C_{v,u}$ and $V_u = \{V \in N_q : N_q \in \langle q, u \rangle\}$//normalization of calculated weights
(11) Store in WL;
(12) EndFor
(13) EndFor
(14) Return WL;
(15) End

Algorithm 5: Weight calculator.

Input: user's query (UQ), weighted logs (WL)
Output: prioritized URLs List (PUL)
Begin
(1) PUL = Ø
(2) Read UQ;
(3) T ← Parser (UQ);
(4) For each URL $u \in U$ in WL do
(5) $W_u = 0$//weight of URL u
(6) For each term $t \in T$ do
(7) $W_u = \sum_{t \in T} W_{t,u} * I_{t,u}$
(8) EndFor
(9) EndFor
(10) If $W_u ! = 0$
(11) PUL = PUL $\cup u$
(12) Sort elements of PUL;
(13) Return PUL;
End
In the next section, an example concerning the above-proposed work is presented.

Algorithm 6: Matcher.

The dataset is divided into two subsets, one for training and the other for testing in the proportion of 80 : 20. The training set has been used to build a prediction model while a testing set comprising various query sets has been used to run multiple test cases. A snapshot of the web access logs is displayed in Figure 9.

5.2. Implementation. Initially, access log file is preprocessed to extract the meaningful entries such as queries and the requested URL and removal of stop words is done. Further queries are parsed into N-grams as shown in Figure 10.

In the next step, weights are assigned to the N-grams. Further, weights are normalized, which is the output of the offline phase, as shown in Figure 11.

In the online phase, when the user submits the query to the server, the prefetching module is also used to predict the user's behavior. A list of prioritized URLs has been given by the online phase to be fetched in the cache before the user's request, as shown in Figure 12.

5.3. Performance Evaluation. In literature [33, 36], prediction performance is measured using two primary

Table 3: Sample of preprocessed logs.

URL	Query after removing stop words
http://www.ymcaust.in	Ymca
http://www.amity.edu	Ncr college
http://www.ymcaust.in	Gov college
http://www.galgotias.org	Top university
http://www.gdgoenka.edu	Ncr college
http://www.ymcaust.in	Ymca
http://www.amity.edu	Amity
http://www.gdgoenka.edu	Top university
http://www.galgotias.org	Galgotias
http://www.amity.edu	Best college
http://www.amity.edu	Amity
http://www.ymcaust.in	Ymca
http://www.gdgoenka.edu	Top university
http://www.galgotias.org	Galgotias
http://www.amity.edu	Best college
http://www.amity.edu	Amity
http://www.ymcaust.in	Ymca
http://www.amity.edu	Ncr college
...	...

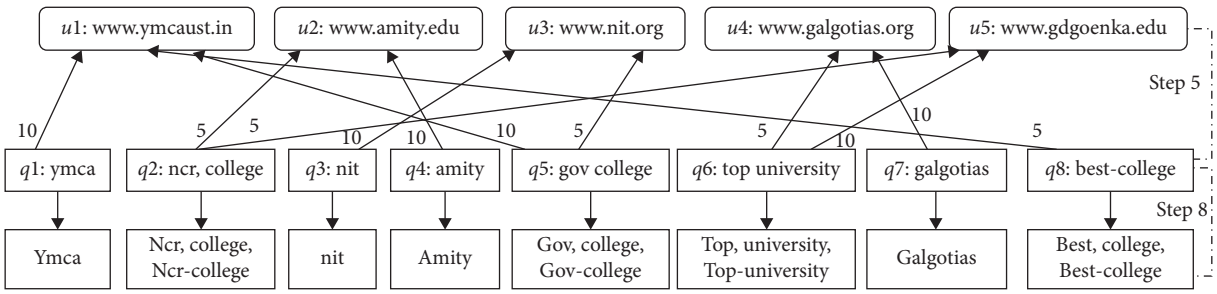


Figure 6: Generation of NC-graph.

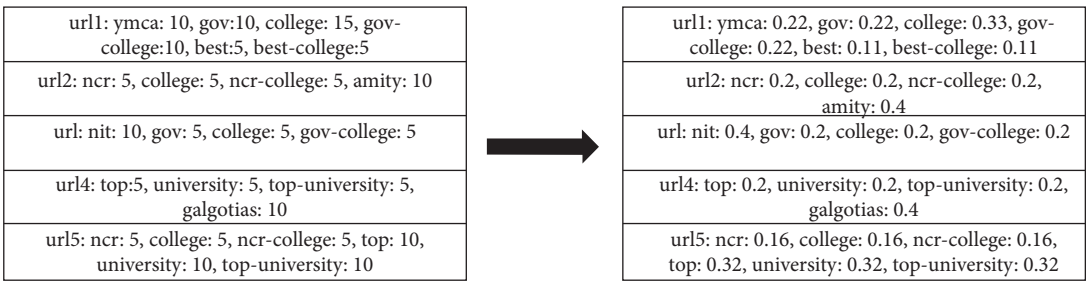


Figure 7: Generation of normalized weights.

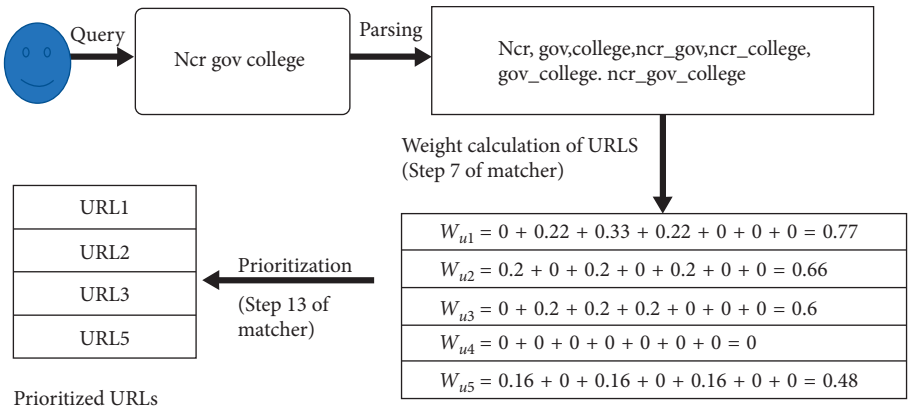


Figure 8: Generation of prioritized URLs based on the users' given query.

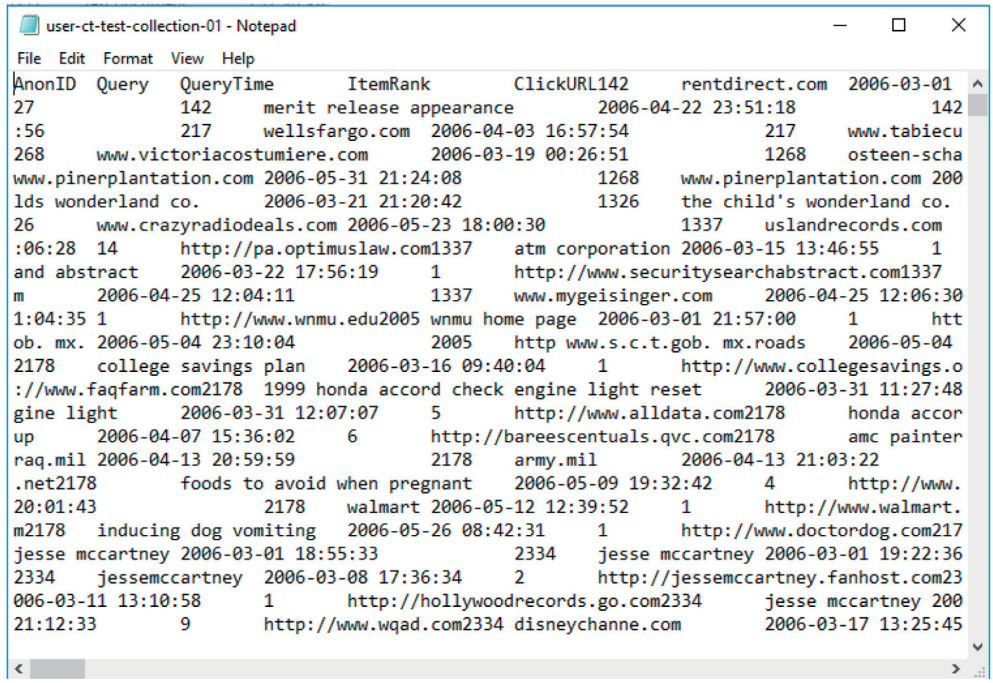


Figure 9: A snapshot of the web access logs.

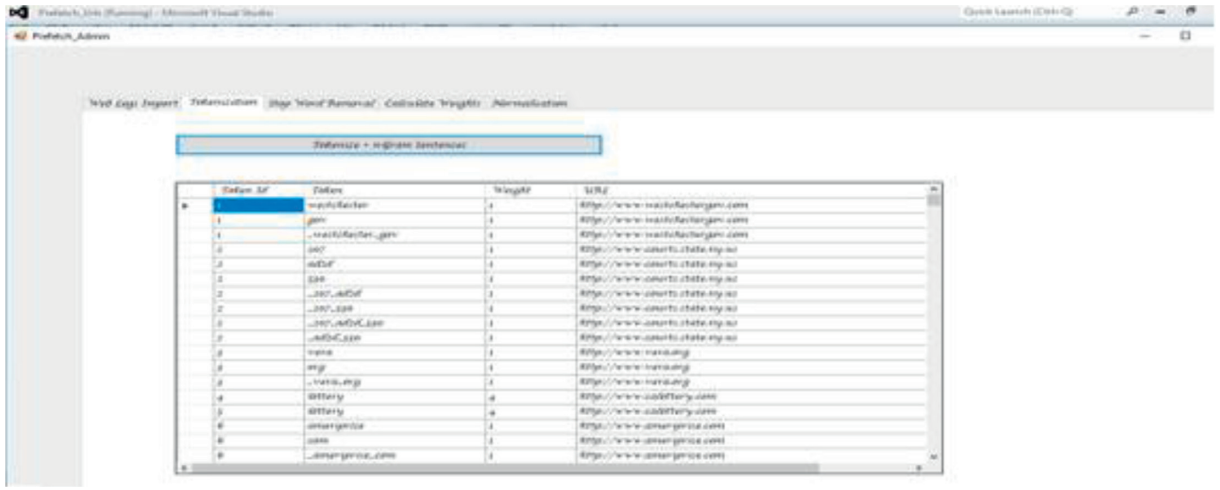


Figure 10: Parsing queries into N-grams.

performance metrics: precision and hit ratio. In our work also, we have used these parameters to measure the accuracy of prediction:

(i) Precision: precision is useful to measure how probable a user will access one of the prefetched pages. Precision is calculated by taking the percentage of the total number of requests found in the cache to the number of predictions.

$$\text{precision} = \frac{\text{total number of requests fetched by the cache}}{\text{total predictions}}. \tag{4}$$

(ii) Hit ratio: hit ratio is useful to measure the probability of the user's request fulfilled by the

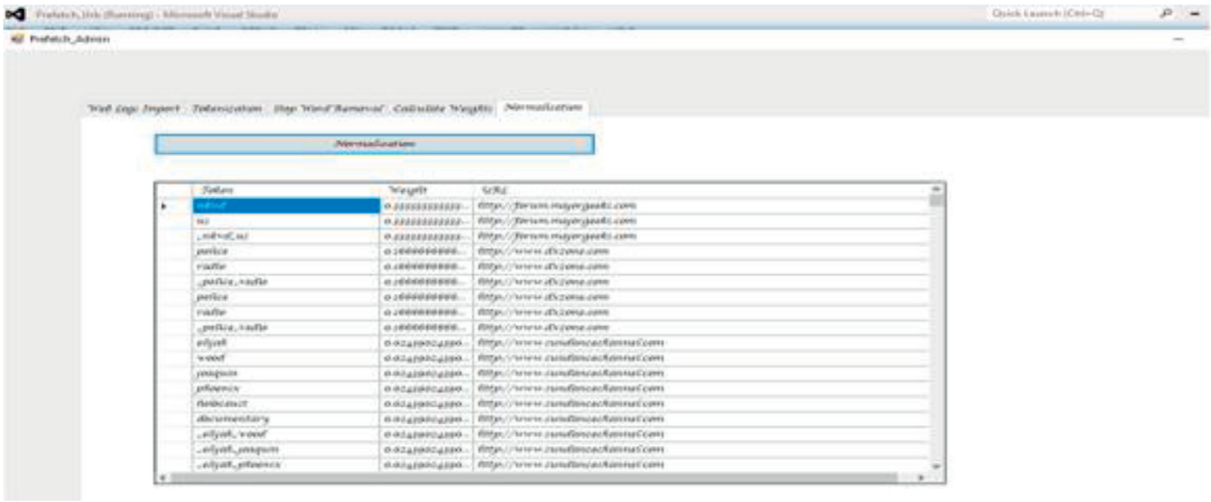


Figure 11: Weighted *N*-grams.

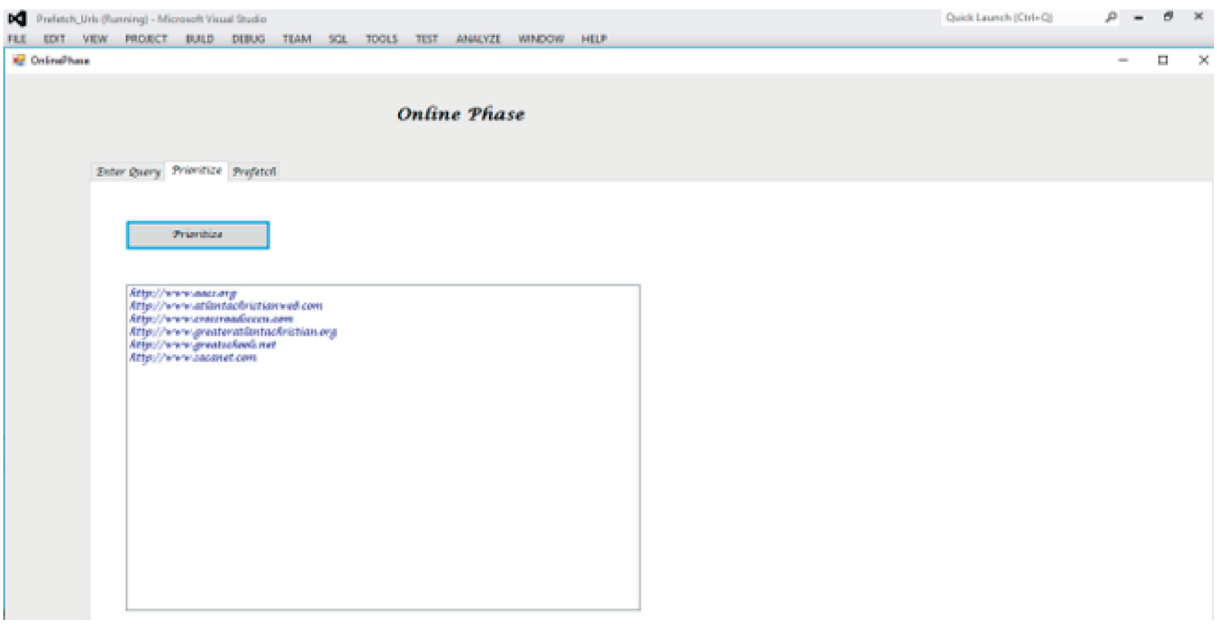


Figure 12: Online phase: prioritized list of URLs.

prefetched pages in the cache. Hit ratio is calculated by taking a percentage of the total number of requests found in the cache to the total number of users' requests.

$$\text{hit ratio} = \frac{\text{total number of requests fetched by cache}}{\text{total users' requests}}. \quad (5)$$

5.3.1. Observation: Impact of N-Grams. This subsection compares the proposed model with *N*-grams against the unigrams approach on the same query sets. Multiple test cases were run by setting up the different thresholds for prefetching. Here, the threshold is a fixed number of pages that are going to be prefetched. On an experimental basis, a broad scale of threshold has been taken. Test cases are discussed as follows:

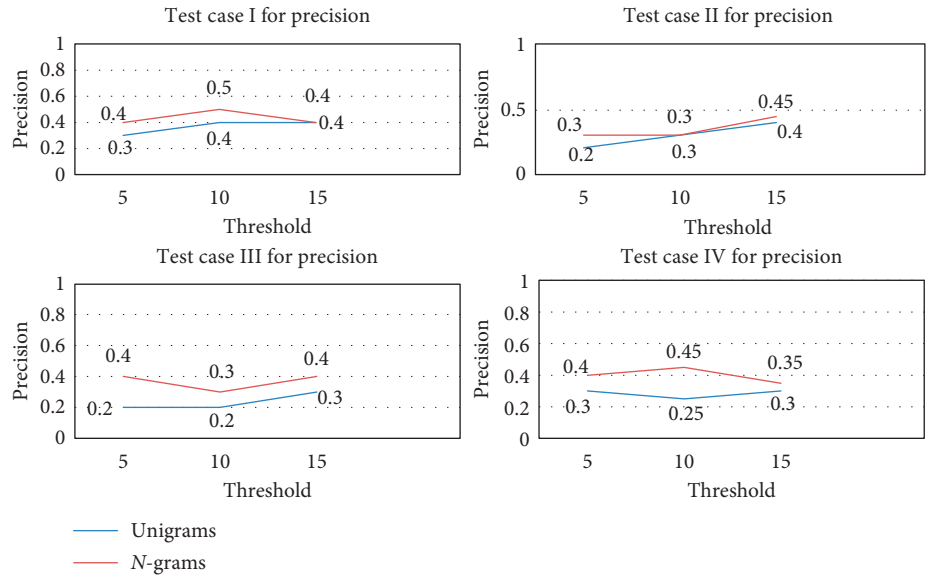


Figure 13: Precision comparison of N-grams and unigrams.

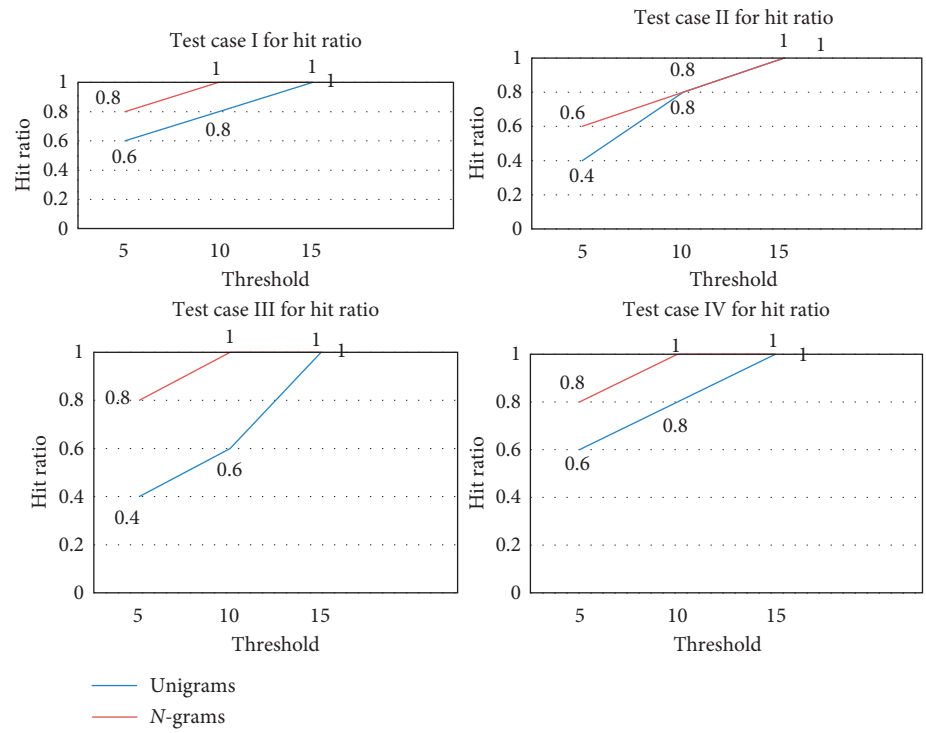


Figure 14: Hit ratio comparison of n-grams and unigrams.

Table 4: Comparison of unigrams and n-grams results for various threshold values.

	Threshold value	Unigram (%)	N-gram (%)	Increase % (%)
Precision	Threshold = 5	25	37	12
	Threshold = 10	28	38	10
	Threshold = 15	35	40	5
Hit ratio	Threshold = 5	50	70	20
	Threshold = 10	70	90	20
	Threshold = 15	100	100	0

Table 5: Comparison of latency.

Average time taken		Reduction (%) in time
Without prefetch	With prefetch	
751	245	**50.6**

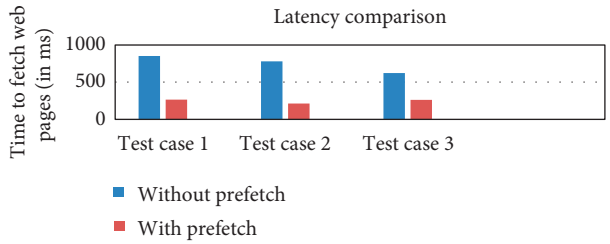


Figure 15: Latency comparison with n-grams prediction model.

Test case I: test the effectiveness of HPM by taking a query having two keywords. Two-keyword-based queries have been extracted from the same AOL logs to run the test case, and an approximate 55000 queries appropriate for this test case were found.

Test case II: test the effectiveness of HPM by taking a query having five keywords. Five-keyword-based queries have been extracted from the same AOL logs to run the test case, and an approximate 65000 queries appropriate for this test case were found.

Test case III: test the effectiveness of HPM by taking a query having eight keywords. Eight-keyword-based queries have been extracted from the same AOL logs to run the test case, and an approximate 50000 queries appropriate for this test case were found.

Test case IV: test the effectiveness of HPM by taking a query having more than ten keywords. Ten-or-more-keyword-based queries have been extracted from the same AOL logs to run the test case, and an approximate 20000 queries appropriate for this test case were found.

All the test cases were run by taking unigrams as well as N-grams of the query. Based on this, precision and hit ratio curves were plotted to evaluate the proposed model, as shown in Figures 13 and 14, respectively.

In general, models with N-grams yield better results than the unigrams in terms of both measures, i.e., precision and hit ratio.

It can be observed from the above graphs that the results of the HPM are much better with an approximately 9% increase on average in precision and about a 13% increase on average in the HIT ratio, as depicted in Table 4. This implies that when the threshold value is less, i.e., the window to fetch the pages for prefetching is small, better precision and hit ratio are achieved in the case of N-grams as compared to unigrams, although when the prefetch threshold increases up to 15, both cases' performance is the same. But the number of prefetches is more in this case, which is not a practical solution. Thus, we can conclude that our system performs better to yield the optimal results in fetching the relevant web pages while consuming less network bandwidth.

5.3.2. Observation: Impact on Latency. A series of test cases comprising the query sets from the testing set of the access logs were run with different inputs, and it is observed that, by using HPM for prefetching, the time taken to fetch the web pages is almost reduced to half of that without prefetching as shown in Table 5. Hence, latency reduction has also been achieved in an impactful manner. The same is shown in Figure 15.

The results of the graph given in Figure 15 are evaluated in Table 5.

5.3.3. Comparison between Web Usage Mining, Web Content Mining, and Hybrid Model. A comparison between these three has been made with various test cases. A series of test cases were run for several types of sessions, i.e., smaller to longer sessions. In our experiments, association rule mining and Markov model-based technique [11] have been used for the WUM technique, and the keyword-based approach [20] has been used for WCM. The proposed model performed well compared to the other two, as shown in Figure 16.

From experiments, it has been concluded that web usage mining and web content mining may perform better in longer user sessions, but in smaller sessions, these techniques do not perform well. Because usage mining-based methods make their predictions based on URLs' sequences, the longer the sequences, the better the results. Similarly, content mining-based strategies learn the user's behavior as they start surfing, and longer sessions provide better learning. However, the proposed hybrid prediction model performs well in smaller as well as longer sessions. From the graphs depicted in Figure 16, we evaluate the results in Table 6.

From the results, it can be summarized that our approach, i.e., hybrid prediction model, clearly provides better results with an approximately 26% increase on average in

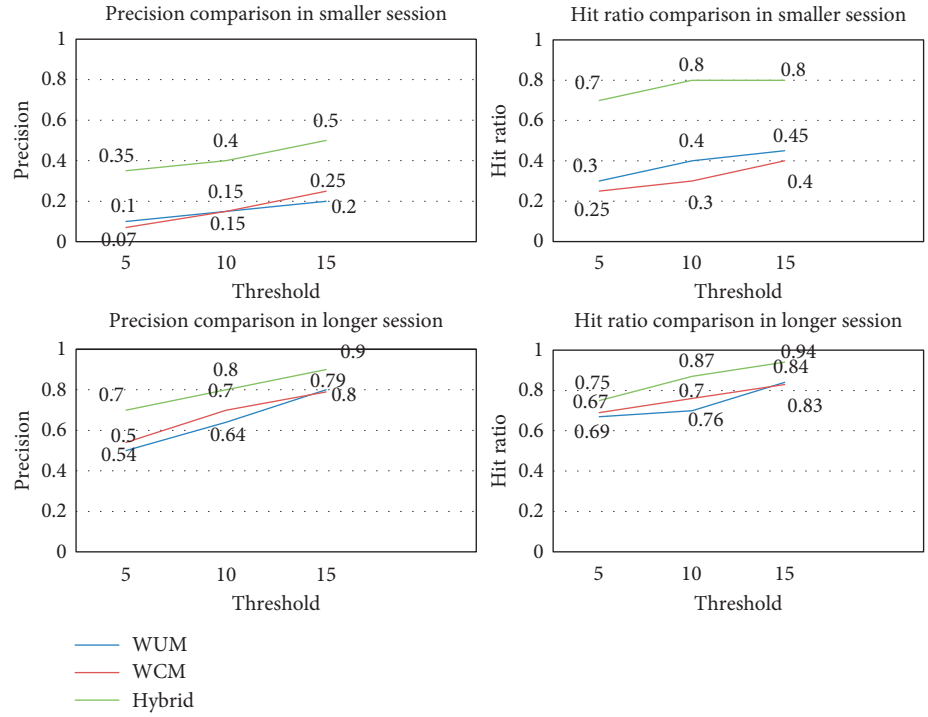


Figure 16: Comparison between WUM, WCM, and hybrid prediction model.

Table 6: Comparison of WUM, WCM, and hybrid approach for precision and hit ratio.

	WUM (%)	Hybrid (%)	Increase (%)	WCM (%)	Hybrid (%)	Increase (%)
Precision	15	41	**26**	15	41	**26**
Hit ratio	41	55	**14**	50	55	**5**

precision and almost an average of roughly 10% increase in HIT ratio.

6. Conclusion and Future Work

Predicting users' behavior in a web application has been a critical issue in the past several years. This work presented a hybrid prediction model that integrates the history-based approach with the content-based approach. History information such as user's accessed web pages is collected from access logs. Our proposed model used Query-URL click-graph derived from the access logs by using queries submitted by the users in the past and corresponding clicked URLs. This Query-URL click-graph is represented in the form of a bipartite graph. *N*-grams are generated by parsing the queries in 3-grams to give more weightage to those *N*-grams which frequently come together and are assigned weights for each URL, and URLs are prioritized by considering the query submitted by the user. The prediction model is efficient and predicts URLs based on content and history. Experimental results have shown a significant improvement in precision of 26% and hit ratio of 10%.

Future work will be devoted to the following:

(i) The prediction model developed so far precisely matches the query terms of the user's interest with the weighted logs. It would be useful to enhance the weighted logs with semantics so that semantics of content could be analyzed to increase the precision and hit ratio further.

(ii) A threshold module will be introduced to dynamically calculate the threshold value based on the server load to optimize the network bandwidth while prefetching.

Data Availability

Data are available upon request to the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] T. M. Kroeger, D. D. E. Long, and J. Mogul, "Exploring the bounds of web latency reduction from caching and prefetching," in *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, pp. 13–22, Monterey, CA, USA, December 1997.

[2] V. Almeida, A. Bestavros, M. Crovella, and A. de Oliveira, "Characterizing reference locality in the WWW," in *Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems*, pp. 92–103, Miami Beach, FL, USA, December 1996.

[3] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, "Changes in web client access patterns: characteristics and caching implications," *World Wide Web: Special Issue on Characterization and Performance Evaluation*, vol. 2, no. 1-2, pp. 15–28, 1999.

[4] S. K. Pal, V. Mitra, and P. Mitra, "Web mining in soft computing framework: relevance, state of the art and future directions," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1163–1177, 2002.

[5] R. Suguna and D. Sharmila, "An overview of web usage mining," *International Journal of Computer Applications*, vol. 39, no. 13, pp. 11–13, 2012.

[6] O. Kumar and P. Bhargavi, "Analysis of web server log by web usage mining for extracting users patterns," *International Journal of Computer Science Engineering and Information Technology Research*, vol. 3, no. 2, pp. 123–136, 2013.

[7] N. Goel, S. Gupta, and C. K. Jha, "Analyzing web logs of an astrological website using key influencers," *International Research Journal*, vol. 5, no. 1, pp. 2–11, 2015.

[8] D. Lee, "Methods for web bandwidth and response time improvement," in *World Wide Web: Beyond the Basics*, M. Abrams, Ed., Prentice Hall, Upper Saddle River, NJ, USA, 1998.

[9] M. Deshpande and G. Karypis, "Selective Markov models for predicting web page accesses," *ACM Transactions on Internet Technology*, vol. 4, no. 2, pp. 163–184, 2004.

[10] D. Kim, N. Adam, I. Im, V. Atluri, M. Bieber, and Y. Yesha, "A clickstream-based collaborative filtering personalization model: towards a better performance," in *Proceedings of the 6th Annual International Workshop on Web Information and Data Management*, pp. 88–95, ACM, Washington, DC, USA, November 2004.

[11] J. Verma, A. Sharma, and G. Amit, "A novel approach to determine the rules for web page prediction using dynamically chosen *K*-order Markov models," *International Journal of Research in Computer and Communication Technology*, vol. 2, no. 12, 2013.

[12] S. G. Oguducu and M. T. Ozsu, "A web page prediction model based on click-stream tree representation of user behavior," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2003.

[13] L. Lu, M. Dunham, and Y. Meng, "Discovery of significant usage patterns from clusters of clickstream data," in *Proceedings of the WebKDD'05*, pp. 139–142, ACM, Chicago, IL, USA, August 2005.

[14] M. A. Awad and I. Khalil, "Prediction of user's web-browsing behavior: application of Markov model," *IEEE Transactions on Systems, Man, And Cybernetics—Part B: Cybernetics*, vol. 42, no. 4, pp. 1131–1142, 2012.

[15] W. Zou, J. Won, J. Ahn, and K. Kang, "Intentionality-related deep learning method in web prefetching," in *Proceedings of the 2019 IEEE 27th International Conference on Network Protocols (ICNP)*, pp. 1-2, Chicago, IL, USA, October 2019.

[16] M. Joo and W. Lee, "WebProfiler: user interaction prediction framework for web applications," *IEEE Access*, vol. 7, pp. 154946–154958, 2019.

[17] J. Martínez-Sugastti, F. Stuardo, and V. González, "Web browsing optimization: a prefetching system based on prediction history," in *Proceedings of the 2017 XLIII Latin American Computer Conference (CLEI)*, pp. 1–10, Cordoba, Argentina, September 2017.

[18] K. M. Veena and R. M. Pai, "Clustering of web users' access patterns using a modified competitive agglomerative algorithm," in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 701–707, Udupi, India, September 2017.

[19] P. Venketesh, "Semantic web prefetching scheme using Naïve Bayes classifier," *International Journal of ComputerScience and Applications*, vol. 7, no. 1, pp. 66–78, 2010.

[20] S. Setia, V. Jyoti, and N. Duhan, "A novel approach for semantic web prefetching using semantic information and semantic association," in *Big Data Analytics*, pp. 471–479, Springer, Singapore, 2018.

[21] T. T. S. Nguyen, H. Y. Lu, and J. Lu, "Webpage recommendation based on web usage and domain knowledge," *IEEE Transactions on Knowledge And Data Engineering*, vol. 26, no. 10, pp. 2574–2587, 2014.

[22] Y. Hu, C. Kang, J. Tang, D. Yin, and Yi Chang, "Large-scale location prediction for web pages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 1902–1915, 2017.

[23] D. Yin, Y. Hu, J. Tang et al., "Ranking relevance in yahoo search," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, August 2016.

[24] Y. Deng and S. Manoharan, "Predicting web accesses using personal history," in *Proceedings of the 2017 IEEE Conference on Open Systems (ICOS)*, pp. 7–12, Miri, Malaysia, November 2017.

[25] P. M. Bharti and T. J. Raval, "Improving web page access prediction using web usage mining and web content mining," in *Proceedings of the 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1268–1273, Coimbatore, India, June 2019.

[26] Z. Chen, Li Tao, J. Wang, L. Wenyin, and W.-Y. Ma, "A unified framework for web link analysis," in *Proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002*, Singapore, December 2002.

[27] B. D. Davison, "Topical locality in the web," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval—SIGIR'00*, Athens, Greece, July 2000.

[28] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," in *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, Francisco, CL, USA, January 1998.

[29] January 2020 http://www.directhit.com.

[30] A. Sheshasaayee and V. Vidyapriya, "A framework for an efficient knowledge mining technique of web page reorganisation using splay tree," *Indian Journal of Science and Technology*, vol. 8, no. 29, pp. 11–15, 2015.

[31] D. A. Vadeyar and H. K. Yogish, "Farthest first clustering in links reorganization," *International Journal of Web and Semantic Technology*, vol. 5, no. 3, pp. 17–21, 2014.

[32] M. B. Thulase and G. T. Raju, "Website reorganization for effective latency reduction through splay trees and concept-based clustering," *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, vol. 325, pp. 173–182, 2015.

[33] C. D. Gracia and S. Sudha, "A case study on memory efficient prediction models for web prefetching," in *Proceedings of the International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pp. 1–6, Pudukkottai, India, February 2016.

[34] S. Kalaivani and K. Shyamala, "A novel technique to preprocess web log data using SQL server management Studio,"

International Journal of Advanced Engineering, Management and Science, vol. 2, no. 7, pp. 973–977, 2016.

[35] January 2020, http://www.researchpipeline.com/mediawiki/index.php?title=AOL_Search_Query_Logs.

[36] C.-Z. Xu and T. I. Ibrahim, "A keyword-based semantic prefetching approach in internet news service," *Journal of IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 5, 2004.

Hindawi
Scientific Programming
Volume 2020, Article ID 8812754, 9 pages
https://doi.org/10.1155/2020/8812754

Research Article

Named Entity Recognition in Chinese Medical Literature Using Pretraining Models

Yu Wang,[1,2] Yining Sun,[1,2] Zuchang Ma,[1] Lisheng Gao,[1] and Yang Xu[1]

[1]*Anhui Province Key Laboratory of Medical Physics and Technology, Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China*
[2]*University of Science and Technology of China, Hefei 230026, China*

Correspondence should be addressed to Yining Sun; ynsun@iim.ac.cn

Received 12 July 2020; Revised 8 August 2020; Accepted 20 August 2020; Published 9 September 2020

Academic Editor: David Ruano-Ordás

The medical literature contains valuable knowledge, such as the clinical symptoms, diagnosis, and treatments of a particular disease. Named Entity Recognition (NER) is the initial step in extracting this knowledge from unstructured text and presenting it as a Knowledge Graph (KG). However, the previous approaches of NER have often suffered from small-scale human-labelled training data. Furthermore, extracting knowledge from Chinese medical literature is a more complex task because there is no segmentation between Chinese characters. Recently, the pretraining models, which obtain representations with the prior semantic knowledge on large-scale unlabelled corpora, have achieved state-of-the-art results for a wide variety of Natural Language Processing (NLP) tasks. However, the capabilities of pretraining models have not been fully exploited, and applications of other pretraining models except BERT in specific domains, such as NER in Chinese medical literature, are also of interest. In this paper, we enhance the performance of NER in Chinese medical literature using pretraining models. First, we propose a method of data augmentation by replacing the words in the training set with synonyms through the Mask Language Model (MLM), which is a pretraining task. Then, we consider NER as the downstream task of the pretraining model and transfer the prior semantic knowledge obtained during pretraining to it. Finally, we conduct experiments to compare the performances of six pretraining models (BERT, BERT-WWM, BERT-WWM-EXT, ERNIE, ERNIE-tiny, and RoBERTa) in recognizing named entities from Chinese medical literature. The effects of feature extraction and fine-tuning, as well as different downstream model structures, are also explored. Experimental results demonstrate that the method of data augmentation we proposed can obtain meaningful improvements in the performance of recognition. Besides, RoBERTa-CRF achieves the highest *F*1-score compared with the previous methods and other pretraining models.

1. Introduction

In recent decades, it has been generally known that the rapid growth of information technology has resulted in huge amounts of information generated and shared in the field of medicine, where the number of published documents, such as articles, books, and technical reports, is increasing exponentially [1]. For example, PubMed houses over 380,000 publications found by just searching the keyword "Diabetes" (Jan. 2009 to Oct. 2019). The medical literature contains valuable knowledge, such as the clinical symptoms, diagnosis, and treatments of a particular disease. However, it is time-consuming and laborious for medical researchers to obtain knowledge from these documents. Thus, it is critical to extract information and knowledge from unstructured medical literature using novel information extraction techniques and present the findings in a visually intuitive Knowledge Graph which supports machine-understandable information about the medicine [2, 3].

Named Entity Recognition (NER) is the fundamental task in Natural Language Processing (NLP). It is also the initial step in extracting valuable knowledge from unstructured text and building a medical Knowledge Graph (KG). As shown in Figure 1, NER aims to recognize entities from unstructured text, and the results of NER may affect

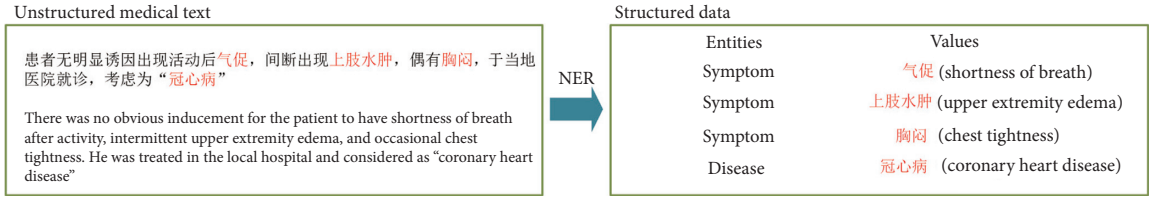


Figure 1: An example of NER.

subsequent knowledge extraction tasks, such as the Relation Extraction (RE). In the early years, researchers used rule-based or dictionary-based methods for NER tasks [4, 5]. However, these methods lack generalization, for they are proposed for particular types of entities. Traditional machine learning and deep learning methods emerging in recent years are also used in NER tasks [6]. Nevertheless, the performance of these methods often suffers from small-scale human-labelled training data, resulting in poor generalization capability, especially for rare words. Moreover, recognizing entities from Chinese documents is a more complex task because there is no segmentation between Chinese characters. Furthermore, in the field of Chinese medical literature, some English symbols, such as the chemical symbols Na and K, may appear in the documents, which makes the NER task more difficult. Therefore, it is of interest to know whether the prior semantic knowledge can be learned from large amounts of unlabelled corpora to improve the performance of NER.

Recently, pretraining models (e.g., BERT and ERNIE) have achieved state-of-the-art (SOTA) results on several NLP tasks. The pretraining models obtain prior semantic knowledge from large-scale unlabelled corpora through pretraining tasks and improve the performance of downstream tasks by transferring this knowledge to them. However, the capabilities of pretraining models have not been fully exploited, and most of the previous works have focused on BERT [7, 8], but applications of other pretraining models in specific domains, such as NER in Chinese medical literature, are also of interest.

In this paper, we enhance the performance of NER in Chinese medical literature using pretraining models. The dataset we used is “A Labelled Chinese Dataset for Diabetes (LCDD),” which contains authoritative Chinese medical literature in recent seven years. The main contributions of this paper can be summarized as follows:

(1) Firstly, we proposed a method of data augmentation based on the Masked Language Model (MLM). Pretraining models will predict the masked words during the procedure of MLM, which can be used for synonym replacement to augment the training set [9]. Considering that there is no segmentation between Chinese characters, we choose ERNIE to conduct this task because it has the entity-level and phrase-level masking strategies.

(2) Secondly, we consider NER as a downstream task of six kinds of pretraining models (BERT, BERT-WWM, BERT-WWM-EXT, ERNIE, ERNIE-tiny, and RoBERTa) and transfer the prior semantic knowledge obtained during pretraining to the downstream task to enhance the performance.

(3) Finally, exhaustive experiments are conducted based on the LCDD dataset. We compare the performance of the NER task on the original and augmented training set. Meanwhile, in addition to comparing the pretraining models with previous methods, we compare the six pretraining models to each other. Moreover, we also explore the performance under different downstream models and two main approaches: feature extraction and fine-tuning. Experimental results demonstrate that the method of data augmentation we proposed can obtain meaningful improvements in the performance of recognition. Besides, RoBERTa-CRF based on the augmented training set with fine-tuning obtains the SOTA result.

2. Related Work

In this section, we will introduce the related works of the Named Entity Recognition, pretraining models, and data augmentation.

2.1. Named Entity Recognition. The Named Entity Recognition aims to identify chunks of text which refer to specific entities of interest, such as drugs, symptoms, treatments, and diseases. Rule-based and dictionary-based approaches had played an important role. For example, Gerner et al. [10] used a dictionary-based approach to identify species names in biomedical literature. Fukuda et al. [11] proposed a rule-based method to extract material names such as proteins from biological documents. However, these methods lack generalization because they need hand-craft rules. Researchers also tried using machine learning methods to recognize entities from unstructured data. He et al. [12] presented a CRF-based approach to recognize drug names in biomedical texts. Wang et al. [13] compared six biomedical NER tools based on the Hidden Markov Model (HMM) and Conditional Random Field (CRF). Nevertheless, machine learning methods need to choose a set of features manually, which is time-consuming and laborious. In recent years, deep learning methods, which can improve the performance of NER without feature engineering, have received increasing attention. For example, Zhu et al. [14] proposed an end-to-end deep learning approach for biomedical NER

tasks which leverages the local contexts via Convolutional Neural Network (CNN). For Recurrent Neural Network (RNN), Chen et al. [15] used a Bidirectional Long Short-Term Memory (BiLSTM) model for the NER from Chinese adverse drug event reports. Chen et al. [16] used dictionary features to help identify rare and unseen clinical named entities. However, deep learning methods still suffer from insufficient training data.

2.2. Pretraining Models. Recently, the pretraining models, which generate representations of words with prior semantic knowledge on large-scale unlabelled corpora, have achieved state-of-the-art results for a wide variety of NLP tasks [17]. Various pretraining models have emerged after Devlin et al. [18] released BERT in 2018. These models consist of multilayer bidirectional Transformer blocks [19]. The main differences among pretraining models lie in the pretraining tasks and pretraining corpora. Table 1 shows the difference in detail. We denote the number of Transformer layers as L, the hidden size as H, and the number of self-attention heads as A. During the procedure of the Next Sentence Prediction (NSP), which is a kind of pretraining task, the pretraining models are trained to predict whether two sentences have a contextual relationship, and the pretraining models can understand the relationship between the sentences in this way.

For the NER task, Devlin et al. [18] first consider NER as a downstream task of BERT for extracting named entities from the news (MSRA-NER). Pires et al. [7] realized zero-shot NER through multilingual BERT. Besides, pretraining models are also used on domain-specific NER, such as biomedicine. For example, Hakala and Pyysalo [8] applied a CRF-based baseline approach and multilingual BERT to the Spanish biomedical NER task. However, the capabilities of pretraining models have not been fully exploited. Furthermore, applications of other pretraining models except BERT in specific domains, such as NER in Chinese medical literature, are also of interest.

2.3. Data Augmentation. A common approach of data augmentation in the area of NLP is synonym replacement [24]. A previous work found synonyms with k-nearest neighbours using Word2Vec [25]. However, the MLM of pretraining models is more suitable for synonym replacement. It is not only because the word representations obtained by the pretraining models contain more abundant semantic knowledge than previous models but also because Word2Vec cannot handle polysemous words. Wu et al. [9] proposed a method of data augmentation based on BERT. However, BERT will mask the Chinese characters, not words, during the procedure of the MLM because there is no segmentation between Chinese characters. Therefore, we perform data augmentation based on ERNIE because it has entity-level and phrase-level masking strategies in the MLM process. The method of data augmentation will be presented in Section 3.1.

3. Methods

3.1. Data Augmentation Using ERNIE. As mentioned earlier, the Masked Language Model (MLM) is intensely suitable for data augmentation. During the procedure of the MLM, a certain portion (e.g., 15%) of words are replaced by a special symbol [MASK], and the pretraining model is trained to predict the masked word. Specifically, for a token sequence $x = \{x_1, \ldots, x_T\}$, the pretraining model first constructs a corrupted sequence $\hat{x}$ by randomly setting a portion of tokens in $\boldsymbol{x}$ to a special symbol [MASK] [26]. The training objective is to reconstruct $\overline{x}$ from $\hat{x}$:

$$\max_{\theta} \log_{p_\theta}(\overline{x} \mid \hat{x}) = \sum_{t=1}^{T} m_t \log_{p_\theta}\left(x_t \mid \hat{x}\right), \tag{1}$$

where $m_t = 1$ indicates that x_t is masked. The whole process is like a *Cloze task* [18]. We repeat the process of MLM using a trained pretraining model. The model is not retrained and is only used to predict masked words. Obviously, the words predicted by the model can be regarded as the synonyms of the masked words. We perform data augmentation based on ERNIE because it has entity-level and phrase-level masking strategies in the MLM process.

A visualization of the process can be seen in Figure 2. ERNIE randomly masks a portion of characters or words in the input sequence by default [21]. It is worth noting that masking the named entities is not appropriate because these entities may be proper nouns or rare words in medical literature, especially the disease and drug entities like "糖尿病 (diabetes)" and "胰岛素 (insulin)" in Figure 2. When ERNIE predicts these entities, the result may not be correct Chinese words because the information of these entities may not be obtained during pretraining. Therefore, we only randomly mask the tokens except for named entities. Furthermore, we input a single sequence that starts with a particular classification token [CLS] and ends with an ending token [SEP], because the context information of sentence pairs is not necessary, which is different from inputting sentence pairs during pretraining [18, 21]. As shown in Figure 2, one sequence input into ERNIE consists of the following four parts:

(1) Token IDs: We use the original vocabulary provided by ERNIE to get the ID number of each token.

(2) Sentence IDs: ERNIE uses this mark to determine the sentences to which the token belongs. As mentioned earlier, we input the single sentence, not a sentence pair. Accordingly, all the sentence ID numbers are 0.

(3) Position IDs: The Transformer cannot obtain position information through self-attention heads, since it contains no recurrence and no convolution [19]. Therefore, the position ID number is injected to get information about the relative or absolute position of the tokens.

(4) Segmentation IDs: The segmentation IDs represent the segmentation information. Specifically, "0" denotes the beginning of a word, and "1" does not

Table 1: Parameters, pretraining tasks, and corpora of pretraining models.

Pretraining model	L	H	A	Pretraining task	Pretraining corpora
BERT [18]	12	768	12	Masked Language Model, NSP	Books Corpus, Wikipedia
BERT-WWM [20]	12	768	12	Whole Word Masking, NSP	Wikipedia
BERT-WWM-EXT [20]	12	768	12	Whole Word Masking, NSP	General data (Baike, News, and QA), Wikipedia
ERNIE [21]	12	768	12	Phrase-level and entity-level masking, NSP	Chinese Wikipedia, Baidu Baike, News, and Tieba
ERNIE-tiny [22]	3	1024	12	Phrase-level and entity-level masking, NSP	Chinese Wikipedia, Baidu Baike, News, and Tieba
RoBERTa [23]	12	768	12	Dynamic masking	Books Corpus, Wikipedia

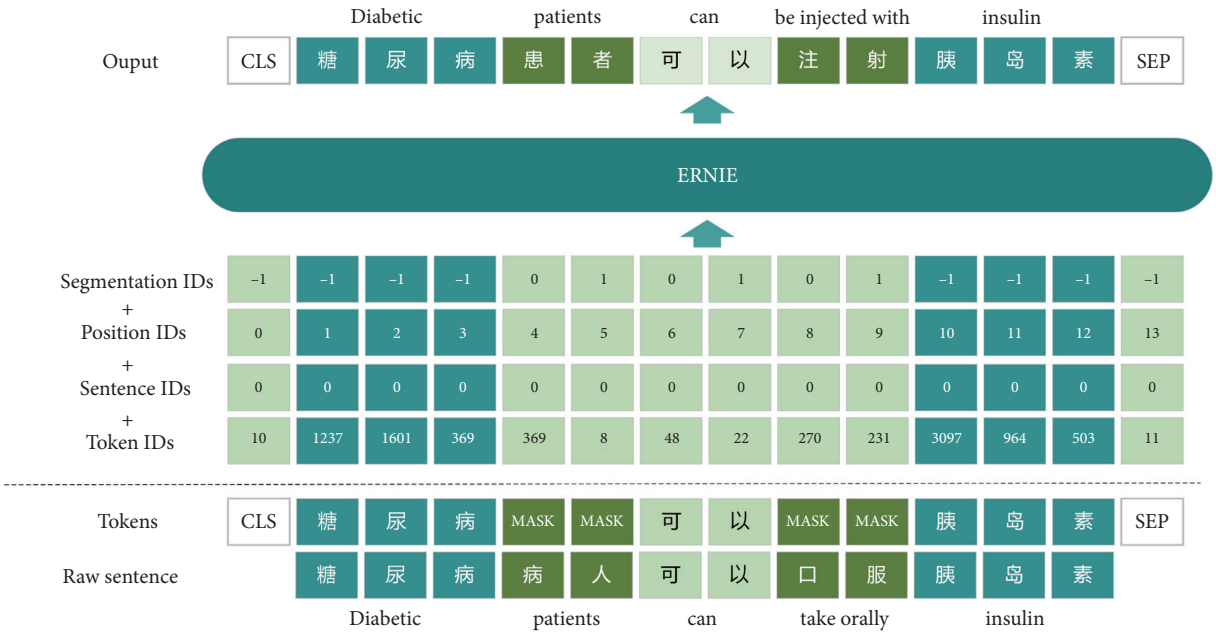


Figure 2: Data augmentation using ERNIE.

denote the beginning. Moreover, we assign "−1" to the corresponding position of [CLS], [SEP], and named entities. ERNIE will not mask the token where the segmentation ID equals "−1." We use THULAC (http://thulac.thunlp.org/) for word segmentation [27].

As can be seen in Figure 2, "病人 (patients)" and "口服 (take orally)" in the raw sentence are replaced by "患者 (patients)" and "注射 (be injected with)," respectively. These two groups of words are synonyms in Chinese. We perform the above operation on all samples in the training set to obtain the dataset D'. Finally, we combine the dataset D' generated by ERNIE with the original training data D to get the augmented training data D_{aug}.

3.2. Named Entity Recognition Using Pretraining Models. We consider NER in medical literature as the downstream task of the pretraining model. As the pretraining models are pretrained on large-scale unlabelled corpora, the output of pretraining models can be regarded as the representations of tokens with prior semantic knowledge. The key to using a pretraining model for NER is how to transfer the prior semantic knowledge obtained from the source domain to the target domain (e.g., Chinese medical literature NER in this paper). There are two main approaches to transfer the prior semantic knowledge to the downstream tasks: feature extraction and fine-tuning [28]. For feature extraction, the parameters of pretraining models are fixed and only the parameters in downstream models are trained through the downstream task. The pretraining models are regarded as the feature extractors and output the representations of tokens with prior semantic knowledge in the source domain. The representations, which are higher-level and more abstract features, will be input into the downstream task. On the other hand, for fine-tuning, all the parameters of pretraining models and downstream models are trained through the downstream task. The pretraining models will learn the semantic knowledge of the target domain from the training data of the downstream tasks. These two approaches are illustrated by Figure 3, where areas marked by blue squares indicate that the parameters of the corresponding models are trained through the downstream task.

For the structure of downstream model, we test the following three common modules: Full Connection (FC), LSTM, and CRF. As shown in Figure 3, the LSTM and CRF are optional. The performance of different modules will be shown in the fourth section.

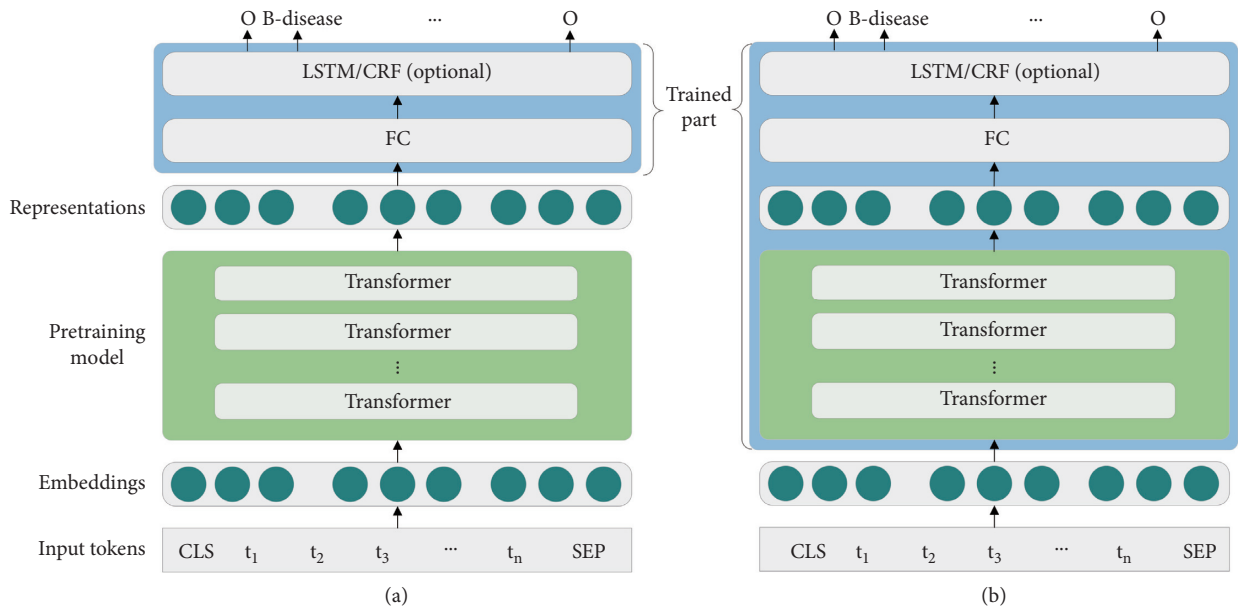


Figure 3: Two main approaches: (a) feature extraction and (b) fine-tuning.

4. Experiments and Results

In this section, we will introduce the dataset for the NER task and show the results. The experiments were performed with PaddlePaddle, which is a framework of deep learning. For hardware, we used an eight-core CPU and a V100 GPU.

4.1. Dataset. The dataset we used is "A Labelled Chinese Dataset for Diabetes," which is provided by Alibaba Cloud [29]. This dataset comes from the authoritative Chinese diabetes journals in recent seven years, from which the literature related to basic research, clinical research, drug usage, diagnosis, and treatment methods are selected. The dataset covers the latest research hotspots on diabetes and is labelled by professionals with a medical background. We divided this dataset into training set, development set, and test set within the ratio of 6 : 2 : 2. The details of the labels are given in Table 2.

4.2. Experiment Settings. We tested the performance of NER from the following three aspects:

(1) Using the method of data augmentation we proposed

(2) Using pretraining models and common deep learning models like the BiLSTM

(3) Using downstream models with different structures

Firstly, we tested the performance using the original dataset and the augmented dataset. Then, the performance of pretraining models, including the BERT series, ERNIE, and RoBERTa, was compared with common deep learning models, such as BiLSTM. Finally, we compared the performance when the downstream model is the LSTM or CRF. For the pretraining models, the parameters were established based on the pretrained parameters provided by their authors. For the downstream models, the weights were established using Xavier initialization, while the biases were initialized as 0.

The hyperparameters are set up based on trial and error. We evaluated the performance at every 1000 steps on the development set, and the experiment would be terminated prematurely once the loss no longer drops. The final selection of the hyperparameters would be the best on the development set. All the hyperparameters involved are listed in Table 3.

For the evaluation, we introduced the precision, recall, and $F1$-score. The precision value refers to the ratio of correct entities to predicted entities. The recall value is the proportion of the entities in the test set which are correctly predicted. The $F1$-score is calculated according to the following formulation:

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (2)$$

It can be seen that the $F1$-score is the harmonic mean of the precision and recall, which can comprehensively reflect the performance of the model on NER tasks. We use P, R, and F to represent precision, recall, and $F1$-score, respectively.

4.3. Results. Firstly, we tested the effects of data augmentation method we proposed. The augmented dataset is obtained through the MLM of ERNIE as described in the third section. We used three pretraining models (BERT, ERNIE, and ERNIE-tiny) based on the original dataset and augmented dataset, respectively. The parameters of pretraining models are updated through fine-tuning. The downstream model is a single-layer FC without the CRF or LSTM. The results are shown in Table 4.

The performance of NER in Chinese medical literature can be improved when using the augmented dataset,

Table 2: Statistics of "A Labelled Chinese Dataset for Diabetes."

	Training set	Development set	Test set
Disease related			
Disease	25197	8399	8399
Reason	2849	950	950
Symptom	3166	1055	1056
Test	28819	9606	9606
Test value	6402	2134	2134
Therapy related			
Drug	9946	3315	3315
Frequency	309	103	103
Amount	871	290	290
Method	606	202	203
Treatment	896	298	299
Operation	493	164	164
Side effect	1052	351	350
Common entities			
Duration	6543	2181	2180
Anatomy	16866	5622	5622
Level	1333	446	448
Total	105348	35116	35119

Table 3: Hyperparameters.

Parameters	Values
Learning rate	$5e-5$
Batch size	32
Weight decay	0.01
Epoch	6
Optimizer	Adam optimizer

and the $F1$-score can be increased by approximately 0.14% on average. The subsequent experiments are all based on the augmented dataset.

Then, we compared the performance when using pretraining models and common deep learning models. The results are shown in Table 5. The parameters of pretraining models are also updated during fine-tuning, and the downstream model is a single-layer FC without the CRF or LSTM, too. As we can see from Table 5, using pretraining models can obtain meaningful improvements in the performance of NER. Among pretraining models, the $F1$-score of ERNIE-tiny is the lowest, at only 89.466%. In contrast, RoBERTa obtained the highest $F1$-score with 91.209%. Moreover, the performance of BERT series models (BERT, BERT-WWM, and BERT-WWM-EXT) is relatively higher than that of ERNIE.

Furthermore, we also compared the two main approaches transferring prior semantic knowledge to the NER task: feature extraction and fine-tuning. For feature extraction, we fixed the parameters of pretraining models. On the contrary, the parameters of pretraining models were trainable and can be updated during fine-tuning based on the training set. The downstream model structure is also a single-layer FC without the CRF or LSTM. The results shown in Table 6 indicate that the $F1$-score can be slightly increased through fine-tuning.

Table 4: Recognition results of original dataset and augmented dataset.

	P (%)	R (%)	F (%)
Models			
BERT	90.778	90.674	90.726
ERNIE	90.488	90.354	90.421
ERNIE-tiny	89.361	89.162	89.261
Models (augmented dataset)			
BERT	90.968	91.048	91.008
ERNIE	90.659	90.527	90.593
ERNIE-tiny	89.466	89.348	89.407

Table 5: Recognition results of pretraining models and deep learning models.

	P (%)	R (%)	F (%)
Deep learning models			
BiGRU	89.431	81.842	85.443
BiGRU-CRF	88.463	84.332	86.341
BiLSTM	89.511	82.251	85.700
BiLSTM-CRF	89.113	84.983	86.992
Pretraining models			
BERT	90.968	91.048	91.008
BERT-WWM	91.023	91.108	91.065
BERT-WWM-EXT	91.059	90.996	91.027
ERNIE	90.659	90.527	90.593
ERNIE-tiny	89.466	89.348	89.407
RoBERTa	91.164	91.254	91.209

Finally, we also tested the performance of different downstream model structures. RoBERTa was used as the pretraining model in this test. For the downstream model, we tested the FC, CRF, LSTM-CRF, and BiLSTM-CRF, respectively. For LSTM-CRF and BiLSTM-CRF, the dimension of the hidden layer was 128. It can be found from Table 7 that the performance of recognition reduced when a fairly complex model was used as the downstream model.

5. Discussion

In this section, we will discuss the experimental results in detail.

5.1. Data Augmentation. Results also show that the augmentation method we proposed can increase the $F1$-score by approximately 0.14% on average. Although the improvement is not significant, the result is meaningful for it demonstrates that the data augmentation using ERNIE is feasible. As mentioned in Section 2.3, BERT will mask the Chinese characters, not words, during the procedure of the MLM because there is no segmentation between Chinese characters, and the results may not be grammatically correct Chinese sentences. However, the MLM of ERNIE can replace a portion of Chinese phrases or words with synonyms. The semantics of the new Chinese sentences generated by ERNIE are similar to those of the original sentences, and they are combined as the augmented dataset. We do not mask the named entities in light of these entities which may

Table 6: Recognition results of feature extraction and fine-tuning.

Pretraining models	P (%)	R (%)	F (%)
BERT (feature extraction)	90.881	90.983	90.932
BERT (fine-tuning)	90.968	91.048	91.008
ERNIE (feature extraction)	90.519	90.639	90.579
ERNIE (fine-tuning)	90.659	90.527	90.593
RoBERTa (feature extraction)	91.109	**91.275**	91.192
RoBERTa (fine-tuning)	**91.164**	91.254	**91.209**

Values in bold represent the maximum values.

Table 7: Recognition results of different downstream model structures.

Pretraining models	P (%)	R (%)	F (%)
RoBERTa-FC	91.164	91.254	91.209
RoBERTa-CRF	91.187	91.358	91.270
RoBERTa-LSTM-CRF	90.615	90.784	90.697
RoBERTa-BiLSTM-CRF	90.820	90.911	90.650

be proper nouns or rare words in the field of medical literature. The results also demonstrate that the augmentation method we proposed is meaningful and feasible.

5.2. Comparison of Pretraining Models with Common Deep Learning Methods. Obviously, using pretraining models can obtain meaningful improvements in the performance of NER. The pretraining models have learned abundant prior semantic knowledge from the pretraining corpora (e.g., Chinese Wikipedia and Baidu News) [20, 21]. Pretraining corpora can also be regarded as the "source domain." When conducting the NER task, the prior semantic knowledge will be transferred to the downstream task, which can also be known as the "target domain." The whole process can be regarded as transfer learning. Task-specific semantic knowledge contained in the target domain will be obtained during fine-tuning.

On the contrary, the common deep learning models can only learn knowledge from the training set, also known as the target domain. The training process is done from scratch on the target domain, whether it is the baseline model (BiLSTM-CRF) or other deep learning models. Therefore, these models can only learn the knowledge in the target domain from the training set. The experimental results also indicated that using pretraining models can get a meaningful increase in the $F1$-score by at least 3%.

5.3. Comparison between Pretraining Models. We also compared the performances of the six most common pretraining models for NER in Chinese medical literature: BERT, BERT-WWM, BERT-WWM-EXT, ERNIE, ERNIE-tiny, and RoBERTa. First of all, it is shown that the deeper the layer, the better the performance for the pretraining models with similar pretraining tasks and the same pretraining corpus, such as ERNIE and ERNIE-tiny. ERNIE has twelve Transformer layers, but ERNIE-tiny only has three Transformer layers. Although ERNIE-tiny increases the number of hidden units and optimizes the pretraining task with continual pretraining [30], three Transformer layers cannot extract semantic knowledge well. The $F1$-score of ERNIE-tiny is the lowest among all the pretraining models.

Secondly, for pretraining models with the same model structure, RoBERTa obtains the highest $F1$-score. From the perspective of the pretraining task, RoBERTa removes the sentence-level pretraining task because Liu et al. [23] found that removing the NSP loss in BERT can slightly improve the performance of downstream tasks. For the NER in Chinese medical literature, the pretraining models do not need to learn sentence-level semantic knowledge during pretraining, because the inputs are all individual sentences, not sentence pairs. The NSP and Dialogue Language Model (DLM) of BERT and ERNIE are designed to improve the performance of specific downstream tasks, such as SQuAD 1.1, which requires reasoning about the relationship between sentence pairs. Moreover, as mentioned before, RoBERTa can acquire richer semantic representations with a dynamic masking strategy [23]. In contrast, BERT and ERNIE use static masking strategy in every pretraining epoch. Therefore, their performance is slightly lower than that of RoBERTa.

Finally, different pretraining corpora will affect the performance of NER in Chinese medical literature for pretraining models with the same pretraining tasks and the same model structures, such as BERT-WWM and BERT-WWM-EXT. The pretraining corpus of BERT-WWM is the Chinese Wikipedia, while the pretraining corpus of BERT-WWM-EXT includes not only the Chinese Wikipedia but also News and Q&A [20]. The training dataset we used contains formal scientific literature, and the pretraining corpus of BERT-WWM is closer to it. The results in Table 5 demonstrate that the $F1$-score of BERT-WWM is slightly higher than that of BERT-WWM-EXT.

5.4. Comparison of Feature Extraction and Fine-Tuning Approaches. As shown in Table 6, the $F1$-score can be slightly increased through fine-tuning. This phenomenon may indicate that the pretraining models can obtain semantic knowledge from the target domain during fine-tuning. In other words, the representations outputs from the pretraining models are not adapted to the specific NER task well when the pretraining models are only used as a feature extractor, because the task-specific representations cannot be obtained in this case. Thus, general-purpose representations can be obtained through fine-tuning. However, considering that the improvement is not significant and the feature extraction is computationally cheaper than fine-tuning, the transfer method should be selected in light of specific conditions in practice.

5.5. Comparison of Different Downstream Model Structures. According to the results in Table 7, RoBERTa-CRF obtained the SOTA results. For the NER task, there are strong dependencies across labels. For example, the I-Drug label must follow the B-Drug label. As a probability model, the CRF can output the predicted sequence according to the above rules.

Therefore, the performance of RoBERTa-CRF is better than that of RoBERTa-FC with only one FC layer.

The experimental results in Table 7 also demonstrate that adding the LSTM after RoBERTa does not improve the performance of recognition. The reason is that, on the one hand, the multiheaded self-attention network in the pretraining model has extracted the abstract representations of input tokens well. Therefore, it is not necessary to add the LSTM to extract more abstract representations. On the other hand, a more complex network structure may cause overfitting, which will reduce the performance of recognition.

6. Conclusion

In this paper, we utilize the pretraining models to recognize the named entity in Chinese medical literature, which is the key step in building the medical Knowledge Graph. First of all, we propose a method of data augmentation based on the MLM of ERNIE. A portion of characters and phrases are replaced by synonyms except for the named entities in light of the fact that the named entities may be proper nouns or rare words in the field of medicine. Moreover, we consider NER as a downstream task of the pretraining models and transfer the prior semantic knowledge obtained during pretraining to it.

The results of experiments demonstrate that not only can the data augmentation method we proposed improve the performance of recognition, but also using pretraining models can obtain a meaningful improvement compared with the common deep learning models. Furthermore, for NER in Chinese medical literature, the $F1$-score can be slightly increased through fine-tuning, and using a more complex downstream model will reduce the performance of recognition. For the future work, we will attempt to carry out experiments with a dataset labelled by ourselves and conduct Relation Extraction based on the entities recognized in Chinese medical literature.

Data Availability

The dataset we used is "A Labelled Chinese Dataset for Diabetes," which can be downloaded from the Tianchi network (https://tianchi.aliyun.com/dataset/dataDetail?dataId=22288).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to thank the AIStudio platform for providing free computing resources. This work was partially supported by the major special project of Anhui Science and Technology Department under Grant 18030801133 and Science and Technology Service Network Initiative under Grant KFJ-STS-ZDTP-079.

References

[1] D. Campos, S. Matos, and J. L. Oliveira, "Biomedical named entity recognition: a survey of machine-learning tools," *Theory and Applications for Advanced Text Mining*, vol. 11, pp. 175–195, 2012.

[2] J. Du and X. Li, "A knowledge graph of combined drug therapies using semantic predications from biomedical literature: algorithm development," *JMIR Medical Informatics*, vol. 8, no. 4, 2020.

[3] N. Boudjellal, H. Zhang, A. Khan, and A. Ahmad, "Biomedical relation extraction using distant supervision," *Scientific Programming*, vol. 2020, no. 9, Article ID 8893749, 2020.

[4] C. Friedman, P. O. Alderson, J. H. M. Austin, J. J. Cimino, and S. B. Johnson, "A general natural-language text processor for clinical radiology," *Journal of the American Medical Informatics Association*, vol. 1, no. 2, pp. 161–174, 1994.

[5] R. Gaizauskas, G. Demetriou, and K. Humphreys, "Term recognition and classification in biological science journal articles," in *Proceedings of the Computational Terminology for Medical and Biological Applications Workshop of the 2nd International Conference on NLP*, 2000.

[6] G. Zhou and J. Su, "Named entity recognition using an HMM-based chunk tagger," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pp. 473–480, Philadelphia, PA, USA, 2002.

[7] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?" in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4996–5001, Florence, Italy, 2019.

[8] K. Hakala and S. Pyysalo, "Biomedical named entity recognition with multilingual BERT," in *Proceedings of the 5th Work-shop on BioNLP Open Shared Tasks*, pp. 56–61, Hong Kong, China, 2019.

[9] X. Wu, S. Lv, L. Zang, J. Han, and S. Hu, "Conditional BERT contextual augmentation," *Lecture Notes in Computer Science*, vol. 11539, pp. 84–95, 2019.

[10] M. Gerner, G. Nenadic, and C. M. Bergman, "Linnaeus: a species name identification system for biomedical literature," *BMC Bioinformatics*, vol. 11, no. 1, 2010.

[11] K. Fukuda, T. Tsunoda, A. Tamura, T. Takagi et al., "Toward information extraction: identifying protein names from biological papers," in *Proceedings of the Pacific Symposium on Biocomputing*, pp. 707–718, Maui, HI, USA, 1998.

[12] L. He, Z. Yang, H. Lin, and Y. Li, "Drug name recognition in biomedical texts: a machine-learning-based method," *Drug Discovery Today*, vol. 19, no. 5, pp. 610–617, 2014.

[13] X. Wang, C. Yang, and R. Guan, "A comparative study for biomedical named entity recognition," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 3, pp. 373–382, 2018.

[14] Q. Zhu, X. Li, A. Conesa, and C. Pereira, "Gram-CNN: a deep learning approach with local context for named entity recognition in biomedical text," *Bioinformatics*, vol. 34, no. 9, pp. 1547–1554, 2018.

[15] Y. Chen, C. Zhou, T. Li et al., "Named entity recognition from Chinese adverse drug event reports with lexical feature based BiLSTM-CRF and tri-training," *Journal of Biomedical Informatics*, vol. 96, 2019.

[16] X. Chen, C. Ouyang, Y. Liu, and Y. Bu, "Improving the named entity recognition of Chinese electronic medical records by combining domain dictionary and rules," *International Journal of Environmental Research and Public Health*, vol. 17, no. 8, 2020.

[17] M. Zaib, Q. Z. Sheng, and W. Emma Zhang, "A short survey of pre-trained language models for conversational AI-a new age in NLP," in *Proceedings of the Australasian Computer Science Week Multiconference*, pp. 1–4, Melbourne, Australia, 2020.
[18] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 4171–4186, Minneapolis, MN, USA, 2019.
[19] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 5998–6008, Long Beach, CA, USA, 2017.
[20] "Chinese-BERT-WWM," https://github.com/ymcui/Chinese-BERT-wwm.
[21] Z. Zhang, X. Han, Z. Liu et al., "ERNIE: enhanced language representation with informative entities," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1441–1451, Florence, Italy, 2019.
[22] "ERNIE-tiny," https://github.com/PaddlePaddle/ERNIE.
[23] "RoBERTa," https://github.com/pytorch/fairseq.
[24] J. Wei and K. Zou, "EDA: easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6382–6388, Hong Kong, China, 2019.
[25] W. Y. Wang and D. Yang, "That's so annoying!!!: a lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviours using # petpeeve tweets," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2557–2563, Lisbon, Portugal, 2015.
[26] Z. Yang, Z. Dai, Y. Yang et al., "Generalized autoregressive pre-training for language understanding," in *Proceedings of the 2019 Advances in Neural Information Processing Systems (NIPS)*, pp. 5754–5764, 2019.
[27] Z. Li and M. Sun, "Punctuation as implicit annotations for Chinese word segmentation," *Computational Linguistics*, vol. 35, no. 4, pp. 505–512, 2009.
[28] M. E. Peters, S. Ruder, and N. A. Smith, "To tune or not to tune? Adapting pre-trained representations to diverse tasks," in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pp. 7–14, Florence, Italy, 2019.
[29] Alibaba cloud labelled Chinese dataset for diabetes, https://tianchi.aliyun.com/dataset/dataDetail?dataId=22288.
[30] Y. Sun, S. Wang, Y. Li et al., "ERNIE 2.0: a continual pre-training framework for language understanding," *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, vol. 34, no. 5, pp. 8968–8975, 2020.