IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 24, 2019, accepted December 8, 2019, date of publication December 18, 2019, date of current version January 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2960659

Extracting Deep Personae Social Relations in Microblog Posts

YAJUN DU[1], FANGHONG SU[2], ANZHENG YANG[1], XIANYONG LI[1], AND YONGQUAN FAN[1]

[1]School of Computer and Software Engineering, Xihua University, Chengdu 610039, China
[2]Sichuan Lewei Technology Company, Ltd., Chengdu 610041, China

Corresponding author: Yajun Du (duyajun@mail.xhu.edu.cn)

This work was supported in part by the National Nature Science Foundation under Grant 61872298, Grant 61532009, and Grant 61472329, and in part by the Sichuan Science and Technology Program under Grant 2018GZ0096 and Grant 2019GFW115.

ABSTRACT Numerous studies have been conducted to extract relationships from different documents. However, extracting relationships from microblog posts is rarely studied. In this paper, we improve a novel kernel-based learning algorithm to mine the personae social relationships from microblog posts by combining the syntax and semantic meanings of the dependency trigram kernels (DTK). To deeply extract the personal social relationships of microblog posts, we define the relation feature words, provide seven rules for extracting these feature words, and propose a rule-based approach that mines these relation feature words from microblog posts. We construct relation feature word dictionaries for different relation types because of the lack of prominent relation features in microblog posts. We propose an algorithm to classify relation feature words by considering two features of the relation feature words, namely, syntax and semantic similarities between relation feature words in microblog posts and by using relation feature word dictionaries. Experimental results show that the average recall, precision, and F-measure of our proposed approach outperforms the original DTK in sentence selection, personae social relation extraction, and personae social relation classification. Finally, the relation graphs of five topics clarify that our proposed approach is effective for extracting personae social relations from microblog posts.

INDEX TERMS Personae relation, relation feature word, dependency trigram kernel, relation classification, knowledge graph, microblog post.

I. INTRODUCTION

The web is an important platform for searching useful information. At present, an increasing number of people are using social media, such as Twitter, Facebook and microblogs, which generates a large quantity of microtext information (such as microposts and videos) every day. One or several microblog posts usually cannot provide useful and valuable information. However, a number of microblog posts on microblog platforms can provide public opinions and important events of the network for the general public and the government. Microblog posts become increasingly complex with time, resulting in the inability of researchers to obtain useful information from historical microblog texts. Therefore, knowledge graph systems are necessary for solving this problem on microblog platforms [1]. Several knowledge graphs, such as Google Knowledge Graph [2], Microsoft Satori [3], DBpedia [4], and Freebase [5], have been developed and used extensively by search engines to enhance their semantic search functions. The basic components of knowledge graphs [6] include the number of entities (persons, things, places, events and topics) and relations among these entities extracted from the web. The information extraction technology, which is very important for these systems, mainly includes three tasks [7], [8]: entity recognition, entity relation extraction, and event detection. Many relation extraction approaches have been successfully applied to long texts, and rarer studies have discovered how to extract the entities and relations from microtexts [9], [10]. Two important international conferences, the Message Understanding Conference (MUC) [13] and Automatic Content Extraction (ACE) [11], guide the relation extraction technologies for documents. The MUC has defined many relation templates to mine different types of relations from documents, such as employee_of, product_of, location_of among organizations. Many methods have been proposed for

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

relation extraction in the corpus of ACE2008. This document defines the six relation types, such as Art (artifact), Gen-Aff (General-affiliation), Org-Aff (Org-affiliation), Part-Whole (Part-to-Whole), Per-Soc (Person-Social), and Phys (Physical).

We classify the relation extraction approaches into three categories: rule-based, feature-based, and kernel-based approaches. They focus on mining relations from some documents, such as web pages and news reports. A large number of research works [14] show that these methods are very successful in dealing with entity and relation extraction in long texts because these sentences in these long texts often have clear semantic meanings, the semantics of words in sentences are almost unambiguous, and the vectors of long texts can not be sparser than short texts. These approaches are ineffective in terms of ambiguous sentence and word semantics, sparse data and so on. Recently, an increasing number of people are using social media. Sina microblogs releases hundreds of millions of microposts every day, generating 50 GB of microtext data. Facebook [12] handles 350 million photos, 4.5 billion ''compliments'', and 10 billion messages a day from around the world. These microblog posts, photos, and messages include rich entities (people names, place names, etc.) and relations (friends, adjacencies, etc.) among them. Additionally, the knowledge graph is a very important tool for developing application systems based on microblogs. The knowledge graph saves and organizes entities and their relations extracted from microblog posts. Therefore, extracting relations from these microblog posts is an urgent problem to be solved in the information retrieval of microblogs. However, the sentences in these texts are incomplete and short, and their semantics are ambiguous. The words in their sentences usually have multiple meanings. These relation extraction approaches cannot function well with microblog applications because microposts with short texts easily cause data sparsity problems. Thus, we focus on personae social relation extraction of microposts in this paper. Our main contributions are as follows:

- We improve a novel kernel-based learning algorithm (denoted as the dependency trigram kernel, NDTK) to mine personae social relations. This algorithm does not rely on entity information to train microblog posts. We divide the sentences in these microblog posts into dependency trigram kernels (DTKs). We combine the syntax and semantic meanings of the DTK to compute the similarity of two sentences.
- We define the relation feature words (FWs) assigning one type of social personae relation between person entities. We propose a rule-based approach to mine these relation FWs for deeply extracting the personae social relation in microblog posts. We identify seven rules for extracting relation FWs. These rules are based on the entity positions and the word semantic roles obtained from the language technology platform cloud (LTP).
- Then, inspired by the system [30], we design learning algorithms to classify the relation FWs. In the algorithm, we consider the relation types among persons and build the relation FW dictionaries for every relation type. For syntax and semantic viewpoints, we compute the types of the relation FWs.

Some difference comparisons of our proposed NDTK with rule-based, feature-based, kernel-based approaches are listed in Table 1.

TABLE 1. Comparisons of NDTK with previous works.

Approaches	TextType	Semantic features	Syntax features
NDTK	short	$\surd$	$\surd$
Rule-Based	long	$\surd$	$\times$
Feature-Based	long	$\times$	$\surd$
Kernel-Based	long	$\surd$	$\times$

The rest of this paper is organized as follows. In section 2, we introduce the DTK of relation extraction and word similarity approaches. In section 3, we propose our NDTK approach to mine personae social relationships from microblog posts. The experimental results are displayed and analyzed in section 4. We conclude with future works in section 5.

II. RELATED WORKS

A. RELATION EXTRACTION APPROACHES

There are many research works on rule-based, feature-based, and kernel-based approaches. We describe them as follows:

- **Rule-based approaches**. These approaches first extract relation rule models by considering the words, phrases, morphologies, and semantic meanings from the document corpus. Then, the relations are retrieved by matching the rule models. Here, we give several typical representations of the approaches in different periods. Brin [15] proposed the dual iterative pattern relation expansion (DIPRE) to extract the relations among authors and documents. They marked the document corpus and constructed relation rule models. Matsuo *et al.* [16] developed the polyphonet system. To mine the relations, the system extracts the common occurrence information among words appearing in web pages using Google search technologies. Then, they proposed relation class models and classified common occurrence information into different relations. Nie *et al.* [17] mined relations based on the specific domains by considering the semantic similarity and dense clusters of the relation rules. Its precision improved by 4% compared with DIPRE. Xu *et al.* [18] and Zhang *et al.* [19] adopted machine learning to train the relation rule models and proposed trigger words of relations to discover relations on the web.
- **Feature-based approaches**. These approaches first retrieve the word and phrase features to form feature vectors for each relation category. They build classifiers with the new relations discovered from the new documents. Kambhatla [20] proposed a maximum entropy model, which combines several text features, such as lexical, syntactic, and semantic features,

to extract 24 relation subtypes in the ACE 2003 corpus. Che *et al.* [21] used a support vector machine (SVM) to train a dataset for relation extraction. Their learning algorithms needed to select features from the ACE 2004 corpus. Xia and Lehong [22] combined sequence, appearance, punctuation, and context features to extract the relations of terms. Finally, they classified these relations by Bayes classifiers. Liu *et al.* [23] proposed the extremity learning machine based on the neuron network algorithm to extract entity relations. They built a concept model to retrieve the efficient space features that included the sentence features and relations among the sentences. Huang *et al.* [24] considered that the space feature vectors of documents have high dimensionality, leading to sparse data vectors. They used document frequency, information quantity, mutual information quantity, and the chi-square test to reduce the dimensionality. Then, they used SVM to mine the personae relations.

- **Kernel-based approaches**. These approaches computed the similarities of two feature vectors in high-dimensionality space. The similarities are some important parameters for constructing the classifiers of the relations. These approaches usually expressed these features of documents, sentences, words, phrases, and semantic meanings using nonlinear methods, such as tree structures. Moreover, the feature vectors contained considerable hidden information about the entity relations. Zelenko *et al.* [25] designed the kernel function method to extract the entity relations from the nonstructured texts. They proposed the kernel functions to compute the similarities of two texts and adopted the similarities to SVM classifiers to mine person-affiliation and organization-location relations. Yu *et al.* [26] proposed a convolution tree kernel-based method to extract Chinese semantic relations. They utilized entity types, subtypes, and mention types to construct unified syntactic and entity semantic trees and evaluated the experimental results on the ACE 2005 Chinese corpus. Zhou *et al.* [27] proposed phrase kernel-based sensitive context information. The method can automatically retrieve the information of the sensitive context trees of sentences. Then, they proposed the convolution tree kernel of the sensitive context information to classify the entity relations. Zhou *et al.* [28] proposed a novel tree kernel-based method. First, they constructed rich semantic relation trees and then proposed a context-sensitive convolution tree kernel for extracting entity relations. The result shows that this method outperformed other state-of-the-art methods on ACE Relation Detection and Characterization (RDC) corpora. Chun *et al.* [29] proposed the mixed kernel function to compute the similarities of two relations. The kernel functions considered the phrase structures in the convolution kernel and the predicates in decision models. Li *et al.* [30] designed a distributed system to extract Chinese entity relations. They constructed six distributed base learners by combining Zhou's convolution tree kernels and entity feature kernels. Then, three communication rules among these learners were proposed to extract the entity relations. The experiments were performed on an ACE RDC2005 Chinese corpus. In conclusion, their objects of study are special corpora containing numerous entity information and features.

B. DTK ALGORITHM

To extract personae social relations from the ACE corpus and Korean news, Choi and Kim [31] proposed the DTK algorithm based on the SVM. They divided the relation extraction process into two phases. In the first phase, sentences that contain relations are selected. In the second phase, the relation names are identified. The DTK algorithm can transform a sentence into some dependency trigrams. Given a sentence S, the dependency tree of S is shown in Fig. 1.

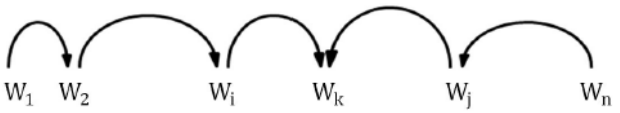


FIGURE 1. A fragment of a sentence dependency tree.

In the sentence, given three words w_i, w_j, and w_k, $w_i \rightarrow w_k$ indicates that word w_i has a dependent relation with w_k, and $w_j \rightarrow w_k$ indicates that word w_j has a dependent relation with w_k. Here, w_k is the common word of the two dependency relations. We denote the $w_i \rightarrow w_k \leftarrow w_j$ as the dependency trigram. We define the dependency trigram set S_T of a sentence into Eq. (1).

$$S_T = S_{T_1} \bigcup S_{T_2}, \tag{1}$$

where

- $S_{T_1} = \{w_i \rightarrow w_k \leftarrow w_j | i < j\}$.
- $S_{T_2} = \{w_l \rightarrow w_k \leftarrow w_r | l < r, \forall w_l, w_r \in child(w_k)\}$.

Given two sentences A and B, $A_T^i \in A_T$ and $B_T^j \in B_T$, let $A_T^i = A_{T_l}^i \rightarrow A_{T_c}^i \leftarrow A_{T_r}^i$ and $B_T^i = B_{T_l}^j \rightarrow B_{T_c}^j \leftarrow B_{T_r}^j$. Choi and Kim fully considered the literal meaning, syntax and part of speech to design the similarity function $s(A_T^i, B_T^j)$ (Eq. (2)) between two dependency trigrams A_T^i and B_T^j.

$$s(A_T^i, B_T^j) = \prod_{k=l,c,r} \sum_{q=1}^{\theta} \alpha_q N_q(A_{T_k}^i, B_{T_k}^j), \tag{2}$$

where

- θ is the number of features (such as word literal meaning, syntax and part of speech) of words in sentences.
- $N_q(A_{T_k}^i, B_{T_k}^j)$ is a binary function, when $A_{T_k}^i = B_{T_k}^j$, then $N_q(A_{T_k}^i, B_{T_k}^j) = 1$, else $N_q(A_{T_k}^i, B_{T_k}^j) = 0$.
- α_q is the weight factor of the qth feature.

According to the dependent trees of sentences, they design kernel functions to select sentences that contain social relations and names of social relations. The number of dependency trigram relations in sentence A is assumed to be less

than that in B. The kernel function (Eq. (3)) is used to select sentences.

$$K(A, B) = \frac{\sum_{i=1}^{n} Max[s(A_T^i, B_T^1), \cdots, s(A_T^i, B_T^m)]}{n}, \quad (3)$$

where

- A, and B represent two sentences. A_T^i, and B_T^j are the dependency trigrams in sentences A and B, respectively.
- n and m are the numbers of dependency trigram relations in sentences A and B.

$K(A, B)$ is a similarity measure function between two dependency trees based on their dependency trigrams. The dependency trigrams are the core components used to calculate the similarity. These components contain various features of sentences, such as word literal meaning, syntax and part of speech. $K(A, B)$ is used to extract the relations in sentence B with case condition A relations as the templates.

The kernel function Eq. (3) considers all dependency trigrams in two sentences to determine whether a new sentence contains relations. The dependency trigrams of a sentence can be expressed in Eq. (4) as follows:

$$S_N = S_{N_1} \bigcup S_{N_2} \bigcup S_{N_3}, \quad (4)$$

where

- $S_{N_1} = \{w_c \rightarrow w_k \leftarrow w_p | \forall\ w_c = child(w_k), \forall\ w_p = parent(w_k)\}$. w_c and w_p are the child nodes and parent nodes of the entity word w_k, respectively.
- $S_{N_2} = \{w_c \rightarrow w_k \leftarrow w_p | \forall\ w_c = parent(w_k), \forall\ w_p = parent(w_k)\}$. w_c and w_p are the child and parent nodes of the entity word w_k, respectively.
- $S_{N_3} = \{w_c \rightarrow w_k \leftarrow w_p | \forall\ w_c = child(w_k), \forall\ w_p = child(w_k)\}$. w_c and w_p are the child and parent nodes of the entity word w_k.

The three kinds of relations indicate that the keywords describing the names of relations usually appear around the entity words. The DTK algorithm finds the relation name using the kernel function Eq. (5) and the dependency trigrams of two different sentences.

$$K(A, B) = s(A_T^i, B_T^j). \quad (5)$$

C. WORD SIMILARITY

The TF-IDF approaches [32] based on the scale text corpora are used widely to determine the statistical similarity between documents. However, these approaches have considerable limits when we use these approaches to calculate the similarity for microblog posts, in which each message contains up to 140 characters. With the help of a knowledge base, several approaches [33]–[35] of computing microblog posts usually expand the sematic meanings of words to reduce some limits. HowNet [36] is a detailed semantic knowledge base. This base, which is represented by a number of words in each composition sememe, is a multidimensional form of words. For example, **Keyboard** is composed of three original composition sememes: Component | 部件, Computer | 电脑, and MusicTool | 乐器; **Relationships** is composed of three original composition sememes: attribute | 属性, relatedness | 亲疏, and human | 人. The original sememe of each level description is unequal. A complex relation exists between the sememes. A special language is needed to describe the relations.

With the original sememe of words, we can calculate the distance or the similarity between two words. The range of distance between two words is $[0, \infty)$. The smaller the similarity is, the farther the distance will be. The distance and the similarity between two words can be established by the following relations:

- The distance between two words is 0, and the similarity is 1.
- The distance between two words is ∞, and the similarity is 0.
- The greater the similarity between two words is, the smaller the distance will be, and vice versa.
- Given two words W_1 and W_2, their similarity can be represented as $Sim(W_1, W_2)$, and the distance between these words is $Dis(W_1, W_2)$ [37], [38]. The relation between the distance and similarity can be represented by Eq. (6).

$$Sim(W_1, W_2) = \frac{\alpha}{Dis(W_1, W_2) + \alpha}, \quad (6)$$

where α is an adjustable parameter, which shows the distance between W_1 and W_2 when their similarity is 0.5.

III. PERSONAE SOCIAL RELATION EXTRACTION

In this section, we introduce our approach for mining personaeśocial relations from microblog posts. Our proposed approach is divided into three parts: mining the personae social relation from the microblog posts; extracting the relation feature words; classifying relation FWs.

A. NDTK APPROACH

The personae social relations in microblog posts are difficult to find by directly using the original DTK algorithm [31]. This limitation is caused by two factors.

- The similarity between two sentences adopts the important position features where the words appear in microblog posts. However, the word positions in microblog posts are usually not obscure.
- Word position in microblog posts cannot be retrieved precisely. Thus, Eq. (3), which depends on $s(A_T^i, B_T^j)$ (it takes on 0 or 1), usually has some low semantic meanings.

To make DTK suitable for handling microblog posts, we improve the similarity function among dependency trigram sets and propose a new function to measure word semantics and syntax weight factors. First, we utilize HowNet to calculate the word semantic similarities in dependency trigrams. Second, we propose (POS, GR) pairs to represent

word syntax similarity $SW(A_T^i, B_T^j)$ (Eq. (7)).

$$SW(A_T^i, B_T^j) = \alpha \times \frac{\sum_{k=l,r} Sim(A_{T_k}^i, B_{T_k}^j)}{2} + \beta \times Sim(A_{T_c}^i, B_{T_c}^j), \quad (7)$$

where

- A_T^i is the ith dependency trigram in the sentence A. B_T^j is the jth dependency trigram in the sentence B.
- $A_{T_r}^i$, $A_{T_l}^i$, and $A_{T_c}^i$ are the left, right, and center words in the ith dependency trigram of the sentence A. $B_{T_r}^j$, $B_{T_l}^j$, and $B_{T_c}^j$ are the left, right, and center words in the jth dependency trigram of sentence B.
- $Sim(w_1, w_2)$ is the word semantic similarity, which is taken from the HowNet.
- α and β (Eq. (8)) are the weights of the similarity of the left, right, and center words in dependency trigrams.

We consider that the center words are the verbs of the relations, while the left and right words are the entity nouns for these relations. The weight β is larger than α because the verb of the relations plays a key role in computing the similarity.

$$\begin{cases} \alpha = \dfrac{NumEntity(A_{T_r}^i \bigcup A_{T_l}^i) + 1}{NumEntity(A_{T_r}^i \bigcup A_{T_l}^i) + IsVerb(A_{T_c}^i) + 2}, \\ \beta = 1 - \alpha, \end{cases} \quad (8)$$

where *NumEntity*() is the number of entity nouns. *IsVerb*() is the number of verbs.

However, the original DTK algorithm considers the part of speech (POS) and grammatical role (GR). This algorithm determines whether the words are the same for the POS and GR of A_T^i and B_T^j. The algorithm is obviously too strict for extracting the personae social relations. Sentence A is '国家主席习近平今天会见美国国务卿克里 (President Jinping Xi meets with U.S. secretary of state Kerrey today)'. Sentence B is '今天美国国务卿克里会见国家主席习近平(U.S. Secretary of state Kerrey meets with president Jinping Xi today)'. '习近平(Jinping Xi)会见(meet)克里(Kerrey)' is a dependency trigram in sentence A. '克里(Kerrey)→会见(meet)←习近平(Jinping Xi)' is a dependency trigram in sentence B. The two dependency trigrams reflect the same relation. However, $N_q(A_{T_l}^i, B_{T_l}^j) = 0$ and $N_q(A_{T_r}^i, B_{T_r}^j) = 0$ in Eq. (2). The POS may be an adjective, verb, or noun. GR indicates that a word belongs to an object, subject, or predicate. The words of the POS and GR are constant in a sentence, and we consider the POS and GR as a whole. Thus, the POS and GR features can be represented as (POS, GR) pairs. We consider that the (POS, GR) contribution of the sentence similarity depends on the frequencies of the left, center, and right words of A_T^i in the corresponding position of B_T^j. For the two features of the (POS, GR), their similarities SC contribute to the dependency trigrams A_T^i and B_T^j are improved by Eq. (8).

$$SC(A_T^i, B_T^j) = \alpha \frac{\sum_{k=l,r} Syn(A_{T_k}^i, B_{T_k}^j)}{2} + \beta Syn(A_{T_c}^i, B_{T_c}^j). \quad (9)$$

- $Syn(X, Y)$ indicates the probability that X appears in Y.
- α and β are the same values in Eq. (7).

The Eqs. (7) and (9) are the semantic similarity and syntax similarity, respectively, for two dependency trigrams. We consider balancing their weights in contributing the sentences using the information entropy and mutual information entropy. The information entropy of words indicates that the words contain the information capacities. The higher the information capacities of the words are, the higher the similarity contributions of the semantic meanings of the words will be. The mutual information entropy of (POS,GR) words indicates their closeness. The higher the mutual information entropy of (POS,GR) is, the closer the (POS,GR) of the words will be, and the larger the similarity contributions of the syntactic features of the words is. Hence, we integrate the semantics and syntax features into a novel similarity of dependency trigram using Eq. (10).

$$s(A_T^i, B_T^j) = \gamma \times SW(A_T^i, B_T^j) + (1 - \gamma) \times SC(A_T^i, B_T^j), \quad (10)$$

where γ is determined by the information entropy of words $E = -\sum_{x \in words} p(x) log_2 p(x)$, and mutual information entropy of (POS, GR) of words $MIE = \sum_{x \in POS} \sum_{y \in GR} p(x, y) log_2 \frac{p(x,y)}{p(x)p(y)}$.

$$\gamma = \frac{E}{E + MIE}. \quad (11)$$

In the detailing implementation of the NDTK approach, the NDTK approach includes two parts: extracting personae social relations and relation feature words and classifying relation FWs.

B. EXTRACT PERSONAE SOCIAL RELATION AND THE RELATION FEATURE WORDS

In microblog posts on the microblogging platform, numerous relationships among persons exist. However, the types of these relations are limited. Using Li's concept [30], we design a basic learner for each type of relation. For convenience of discussion, we consider only four types of personae relations in microblogs, namely, **Work**, **Family**, **Friend**, and **Enemy**. In traditional methods, the FWs are extracted by analyzing a word syntactic structure in a sentence. However, the sentence structure is complex and fuzzy. Thus, traditional methods are complex cases, and inaccuracies exist to determine all correct FWs. Therefore, we utilize NDTK to extract the relation FWs between two entities. These kernel words can represent relation FWs for further classification. For example, the sentence '国家主席习近平今天会见美国国务卿克里 (President Jinping Xi meets with U.S. Secretary of state Kerrey today)',

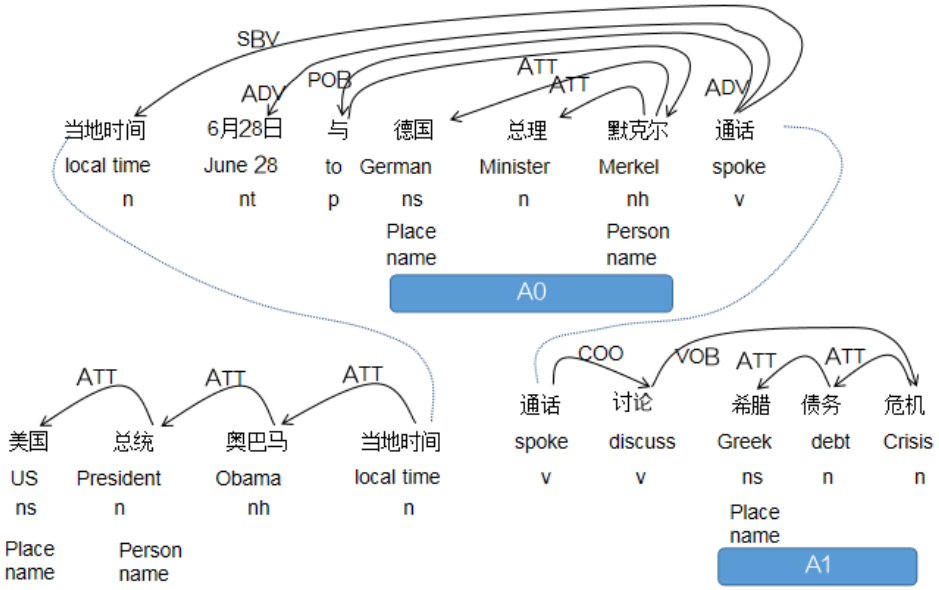


FIGURE 2. The dependence tree of the sentence'US president Obama spoke to German Minister Merkel on June 28, local time, to discuss the greek debt crisis.

we use NDTK to extract the relation FW '会见(meet)' between two person entities '习近平(Jingping Xi)', and '克里(Kerrey)'. We called the word 'meet' an NDTK relation FW. Then, we designed four learners using these FWs for four further relation classifications.

Relation FWs are candidates for depicting personae relations. Emotion analysis and classification [39], [40] utilize an emotion dictionary to construct a model of emotion detection or classification. Inspired by these approaches, we first manually constructed the initial relation dictionary describing the words and then used the relation dictionary to classify them. Finally, we expanded the relation dictionary using the chi-square test, mutual information (MI) and HowNet. To construct the dictionary, we selected the standard relation words from the HowNet dictionary, and each relation type contained approximately 300 words. Given a sentence S, the dependence tree is extracted by utilizing a public platform named LTP [42]. The LTP dependence tree of the sample sentence '美国总统奥巴马当地时间6月28日与德国总理默克尔通话，商讨希腊债务危机(US President Obama spoke to German Minister Merkel on June 28, local time, to discuss the Greek debt crisis)' is shown in Fig 2.

In this dependency tree, 'place name', and 'person name' indicate entity types. We denote $A0$ and $A1$ as semantic roles. $A0$ represents the agent of the actions, and $A1$ represents the receiver of the actions. According to the order of appearance in a sentence, all agents of actions in a sentence can be denoted $\{A_{01}, \cdots, A_{0i}\}$, and all receivers of actions as $\{A_{11}, \cdots, A_{1j}\}$. E_1, and E_2 are two personae entities. In turn, w_1, $\cdots$ and w_n are terms except entities. The positions of entities, which contain a relation in the most likely condition, can be concluded in three situations, and FWs describing the relations can be summarized in seven rules (Table 2).

TABLE 2. Extracting rules of FW of relations.

Position of entity:
$w_1 \cdots w_i E_1 w_x \cdots w_y E_2 w_j \cdots w_n$
Rules of extracting feature words:
Rule1:
If $E_1 \in \{A_{01}, \cdots, A_{0j}\}$ and $E_2 \in \{A_{11}, \cdots, A_{1j}\}$
Then, FW=$IR \bigcup \{w_x \cdots w_y\}$
Rule2:
If $\{E_1, E_2\} \in \{A_{01}, \cdots, A_{0j}\}$
Then, FW=$\{A_{01}, \cdots, A_{0j}\} \bigcup IR - \{E_1, E_2\}$
Rule3:
If $\{E_1, E_2\}$ in $\forall A1_j$
Then FW=$\{A1_j\} \bigcup IR - \{E_1, E_2\}$
Rule4:
Otherwise
FW=$IR \bigcup \{A1_j \mid A1_j = min \ \ Dis(IR, A1)\}$
Position of entity:
$w_1 \cdots w_i E_1 E_2 w_j \cdots w_n$
Rules of extracting feature words:
Rule5:
If IR$\in \{w1 \cdots w_i\}$ then
FW=$IR \bigcup \{A1_j \mid A1_j \in \{w_1 \cdots w_i\}$
$\bigwedge A1_j = min \ \ Dis(E_1, A1)\}$
Rule6:
If IR$\in \{w_j \cdots w_n\}$ then
FW=$\{IR \bigcup A1_j \mid A1_j \in \{w_j \cdots w_n\}$
$\bigwedge A1_j = min \ \ Dis(E_2, A1)\}$
Position of entity:
IR="Is" and child("Is")$\in \{E_1, E_2\}$
Rules of extracting feature words:
Rule7:
FW=$\{N_{A0_i} \bigcup N_{A1_j} \mid N_x$ is noun of x
$\bigwedge A0_i$=min Dis("Is", A0)
$\bigwedge A1_j$=min Dis("Is", A1)$\}$

In Table2, *FW* is a set that contains all words describing relations by using these rules. *IR* represents the person interaction relation word that is extracted by utilizing the NDTK algorithm. $A0$ and $A1$ are semantic roles in the LTP

dependency tree. Function minDis(X,Y) returns word set Y, which is the nearest distance from X to Y in a sentence.

- Rule1: If E_1 is an agent of actions, and E_2 is a receiver of actions; then all words between E_1 and E_2 and all relation words are relation feature words.
- Rule2: If these words exist between E_1 and E_2, and E_1 and E_2 are agents of actions, then all agent of actions words, and all relation words except E_1 and E_2 are relation feature words.
- Rule3: If these words exist between E_1 and E_2, and E_1 and E_2 are receivers of actions, then all words of receivers of actions, and all relation words except E_1 and E_2 are relation feature words.
- Rule4: If these words exist between E_1 and E_2, and E_1 and E_2 are not agents of actions and receivers of actions, then all relation words and words of receivers of actions that maintain a minimum distance with all relation words are relation feature words.
- Rule5: If these words do not exist between E_1 and E_2, and all relation words lie in the left side of agent of actions $E1$, then all relation words and words of receivers of actions that maintain a minimum distance with E_1 are relation feature words.
- Rule6: If these words do not exist between E_1 and E_2, and all relation words lie in the right side of the receiver of actions $E2$, then all relation words and words of receivers of actions that maintain a minimum distance with E_2 are relation feature words.
- Rule7: If the relation word of a sentence is 'is', then the relation feature words include all nouns of the agent of actions and the receiver of actions that maintain a minimum distance with *is*.

For example, there is the sentence "美国总统奥巴马当地时间6月28日与德国总理默克尔通话，商讨希腊债务危机 (US President Obama spoke to German Minister Merkel on June 28, local time, to discuss the Greek debt crisis)" in Fig.2. According to this structure, we use the first rule to get FW. The IR is '通话(spoke)', and it belongs to $A1$. So that FW = {通话(spoke), 商讨(discuss), 希腊(Greek), 债务(debt), 危机(crisis)}.

The detailed description of extracting personae social relations and the relation FWs is shown as follows:

- Step 1. Expanding the relation dictionary by using the chi-square test, mutual information (MI) and HowNet [41].
- Step 2. Dividing the microblog posts into sentences, the sentences in different words, and constructing dependence trees of the different sentences using the LTP tool [42].
- Step 3. Extracting the dependency trigrams from the different sentences using DTK approaches.
- Step 4. Considering the semantic and syntax features of these sentences, we utilize Eqs. (7), (9) and (10) to choose the dependency trigrams.
- Step 5. Extracting the relation FWs between two entities by using these seven rules in Table 2.

C. CLASSIFYING RELATION FWs

In the real world, many relations exist among persons. We manually construct dictionaries with many FWs to describe these relationships. We use $D_j = \{d_1^j, \cdots, d_k^j, \cdots, d_m^j\}$ to represent the jth type of the relation FW dictionaries. The d_k^j is the kth relation FW in the jth type dictionary. $FW_i = \{f_1^i, \cdots, f_q^i, \cdots, f_n^i\}$ is the FW set that describes the relations and is extracted from the *ith* microblog post by using the above rules, and f_q^i is *qth* word describing relation in the ith microblog post. n is the number of relation FWs describing in the ith microblog post. We can construct a similarity matrix between FW_i and D_j. The matrix is shown as follows:

$$M_{m\times n}^{ij} = \begin{pmatrix} Sim(f_1^i, d_1^j) & \cdots & Sim(f_1^i, d_m^j) \\ Sim(f_2^i, d_1^j) & \cdots & Sim(f_2^i, d_m^j) \\ \cdots & \cdots & \cdots \\ Sim(f_n^i, d_1^j) & \cdots & Sim(f_n^i, d_m^j) \end{pmatrix}. \quad (12)$$

Each element of the matrix $Sim(f_q^i, d_k^j)$ represents the semantic similarity [37] between the word d_k^j in the jth dictionary and the relation FWs f_q^i in the ith microblog post. We obtain the greatest value d_q^j of $Sim(f_q^i, d_k^j)$ and the word of the qth row in the similarity matrix. $D_{max}^j = \{d_1^j, d_2^j, \cdots, d_n^j\}$ is the vector corresponding to the maximum word dictionary record. $C(FW_i)$ is the classification of FW_i, where j is relation types. According to the maximum word dictionary record D_{max}^j, we compare the semantic and syntactic feature similarity between the words d_k^j in the dictionary and the relation words f_q^i. We propose a classification algorithm of relation FWs (relation FW classification algorithm, RFWCA). The detailed description of this algorithm is shown as follows:

- In steps 09∼15 of the RFWCA, we first compute for the $|FW_i \bigcap D_j|$. This value indicates that the number of the relation FWs of the microblog post i includes in the relation FW dictionary D_j. Then, we compute the maximum number that belongs to the relation FW dictionaries.
- In steps 16∼20 of the RFWCA, if $| FW_i \bigcap D_j |$ is the only maximum number in $j = \{1, \cdots, n\}$; then, return $C(FW_i)$; and the algorithm stops. Otherwise, it indicates several maximum numbers, and steps 16∼20 calculate the semantic and syntactic similarity between FW_i and D_j.
- In step 23 of the RFWCA, we extract the syntactic features of $FW_i = \{f_1^i, f_2^i, \cdots, f_n^i\}$, such as POS, GR, semantic role, child nodes, and parent nodes.
- In steps 24∼27 of the RFWCA, we obtain the greatest value vectors, replace the corresponding words in $FW_i = \{f_1^i, f_2^i, \cdots, f_n^i\}$ and words in $D_{max}^j = \{d_1^j, \cdots, d_n^j\}$ with syntactic feature words, and then reconstruct a new dependency tree for extracting syntax features.
- Step 28 of the RFWCA compared with FW_i and D_{max}^j. We use Eqs. (13) and (14) to calculate the syntax

Algorithm RFWCA

01 Input
02 $FW_i = \{f_1^i, \cdots, f_q^i, \cdots, f_n^i\}$;
03 $D_j = \{d_1^j, \cdots, d_k^j, \cdots, d_m^j\}$;
04 Output
05 $C(FW_i)$ // Classification of FW_i;
06 Begin
07 $NumMaxD \Leftarrow 0$;
08 $SetofMaxD \Leftarrow \Phi$;
09 for $j = 1$ to m;
10 Begin;
11 $NumD[j] \Leftarrow \mid FW_i \bigcap D_j \mid$;
12 If $NumD[j] \geq NumMaxD$;
13 $NumMaxD \Leftarrow NumD[j]$;
14 $SetofMaxD \Leftarrow SetofMaxD \bigcup j$;
15 End for;
16 If $\mid SetofMaxD \mid = 1$;
17 $C(FW_i) \Leftarrow SetofMaxD.vaule$//get the word of type SetofMaxD;
18 return;
19 Else;
20 $Score \Leftarrow 0$;
21 For each D_j in SetofMaxD;
22 Begin;
23 Extract FW by using DTK;
24 Reconstruct and Expand DTK by D_j;
25 Construct $M_{m\times n}^{ij}(FW_i, D_j)$
26 Get $d_1^j, d_2^j, \cdots, d_n^j$
27 Get $D_{max}^j \Leftarrow \{d_1^j, d_2^j, \cdots, d_n^j\}$
28 $Syn(FW_i, D_{max}^j) = \frac{\sum_{q=1}^{n}\sum_{p=1}^{k} N_j(f_q^i, d_{qp}^j)}{k\times n}$ by Eq. (13) and (14);
//Syntax Similarity between FW_i and dictionary
29 $Sem(FW_i, D_j) \Leftarrow \frac{\sum_{p=1}^{n} a_p^j}{n}$ by Eq. (15)
//Semantics Similarity between FW and dictionary
30 Sc$\Leftarrow Syn(FW_i, D_{max}^j) + Sem(FW_i, D_j)$;
31 If Sc$\geq$Score;
32 Score$\Leftarrow$Sc;
33 $C(FW_i) \Leftarrow C(FW_i) \bigcup SetofMaxD.vaule$;
34 End For
35 return $C(FW_i)$//The result of classification.

similarity between FW_i and D_{max}^j:

$$N_x(f_q^i, d_q^j) = \begin{cases} 1 & if\ the\ same\ features, \\ 0 & otherwise. \end{cases} \quad (13)$$

$$Syn(FW_i, D_{max}^j) = \frac{\sum_{q=1}^{n}\sum_{p=1}^{k} N_j(f_q^i, d_{qp}^j)}{k \times n}, \quad (14)$$

where k represents the number of features. The d_{qp}^j represents the pth of d_q^j, such as POS, GR, children, and parent.

- According to the $M_{m\times n}^{ij}$ dictionary similarity matrix, Step 29 of the RFWCA uses the greatest element of the kth row in the similarity matrix corresponding to d_k^j to calculate the semantic similarity $Sem(FW_i, D_j)$. The equation is shown as follows:

$$Sem(FW_i, D_j) = \frac{\sum_{p=1}^{n} a_p^j}{n}. \quad (15)$$

- Steps 30∼34 of the RFWCA select the maximum score of similarity between words in the relation dictionary and relation feature words as the result of classification. $Score_j$ is computed using the following equation:

$$Score_j = Syn(FW_i, D_{max}^j) + Sem(FW_i, D_j). \quad (16)$$

IV. EXPERIMENTS AND RESULTS

NDKE is developed on the basis of DKE. The rule-based, feature-based, kernel-based approaches with long texts are not comparable with NDKE. Therefore, we only choose DKE as the baseline to compare our proposed NDKE approach. In our experiment, our algorithms run on a computer group of four computers. Every computer includes an Intel(R) Core(TM) i5-3230 M @2.60 GHz, memory of 4.00 GB, hard disk of 1 TB, Windows 7 OS, and distributed system Hadoop. The four (friend, work, family, enemy) initial dictionaries have approximately 1,000 relation feature words. We describe our experimental flow in Fig. 3. It includes the following parts: crawl microblog posts, construct the initial dictionary, construct dependence trees, extract the dependency trigrams, choose the dependency trigrams, extract the relation FWs, classify relation FWs, and construct knowledge graph.

A. DATASETS

To evaluate our proposed NDTK approach for mining deep personae social relations and proposed RFWCA for classifying relation FWs. We experimentally crawled numerous real data about some person topics from the TenCent and Sina microblog platforms. A total of 110,000 original microblog posts (including 100,000 normal microblogs and 13,000 topic microblog posts) were downloaded. We selected 6,968 topic microblog posts and 11,088 normal microblogs as our experimental dataset. We denoted these microblog posts without the five topics to the normal microblog posts. The topics of microblog posts were related to people, so they contained more person entities. Table 3 shows six topics and the numbers of microblog posts in the crawled dataset.

In Table 3, we divided these microblog posts into two parts, namely, topic and normal posts. Topic posts, which approximately described person relations, and contained five topics, such as microblog news, talks, cooperations, politicians, and famous stars. The normal posts without topics were

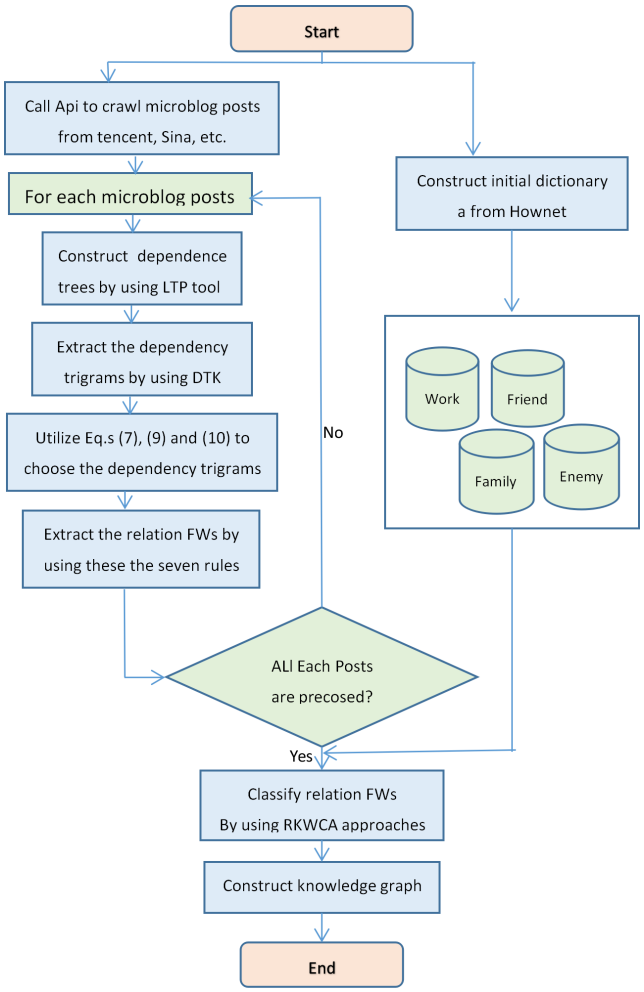


FIGURE 3. The experimental flow.

TABLE 3. Our crawled dataset.

Topic types	Number of posts	Number of sentences NDTK	DTK
Microblog News	1524	1212	1124
Talks	1722	1465	1356
Cooperations	955	812	725
Politicians	1344	1135	1241
Famous stars	1423	1189	1145
Normal Posts	11088	6014	5520

people's individual thoughts in microblogs. The topic posts clearly contained more personae relations. These relations were easier to extract than normal posts.

B. EVALUATION CRITERION

In this paper, we divided the relation extraction processes into two phases. In the first phase, personae social relations were extracted using our proposed NDTK algorithm. In the second phase, the relation feature word sets *FWs* ere extracted and classified into different relation types using the RFWCA.

1) NDTK APPROACH

Three indices P (precision), R (recall) and F (F-measure) [43] were adopted to measure the performances of our proposed NDTK and the original DTK approach.

$$P = \frac{NCE}{NE}, \quad R = \frac{NCE}{NCT}, \; F = \frac{2 * P \times R}{P + R}, \tag{17}$$

where NCE is the number of correct personae social relations extracted from the microblog posts in the test datasets. NE is the number of the social personae relations extracted for microblog posts in the test datasets. NCT is the number of correct personae social relations in the test datasets.

2) RKWCA

In this subsection, we provide several measurement criteria of our proposed rules to extract the relation FWs and classify FWs into different types by RKWCA. We adopt the correct rate FWC of FW to measure the performances of the NDTK algorithm.

$$CKWords_i = \begin{cases} 1 & NumKWords \geq \dfrac{length(FW_i)}{2}, \\ 0 & otherwise, \end{cases} \tag{18}$$

where $CKWords$ represents the correct word set describing the relations. Eq. (18) indicates that if half of the words in the FW_i are correct, then the whole relation FWs are correct.

$$FWC = \frac{\sum_{i=1}^{m} CKWords_i}{m}. \tag{19}$$

We adopt the weighted average precision P_{Avg}, recall R_{Avg} and F-value F_{Avg} to evaluate the performance of the RFWCA.

$$P_{Avg} = \frac{\sum_{j=1}^{n} NumC_j \times P_j}{\sum_{j=1}^{n} NumC_j}, \tag{20}$$

$$R_{Avg} = \frac{\sum_{j=1}^{n} NumC_j \times R_j}{\sum_{j=1}^{n} NumC_j}, \tag{21}$$

$$F_{Avg} = \frac{\sum_{j=1}^{n} NumC_j \times F_j}{\sum_{j=1}^{n} NumC_j}, \tag{22}$$

where P_j is the precision of relation type j, F_j is the F-value of relation type j, R_j is the recall rate of relation type j, and $NumC_j$ is the number of relation instances classified in type j.

C. RESULT ANALYSIS

1) EVALUATION OF NDTK APPROACH

In this subsection, we compare our improved NDTK approach with the original DTK approach based on two aspects, relation sentence selection and personae social relation extraction. We retrieve the Ps, Rs, and Fs of microblog news, talks, cooperations, politicians, famous stars, and normal posts of the NDTK and DTK approaches.

Figs. 4∼6 demonstrate the sentence selection performance of the NDTK and original DTK approaches. In Fig. 4, we discover that the precision *P*s of NDTK's sentence selection is higher than that of the original DTK approach in microblog news, talks, famous stars, and normal posts but lower than that of the DTK approach in politicians and cooperations. This phenomenon may be caused by the smaller number of samples in politicians and cooperations. The average *P*s of the sentence selection of NDTK and the original DTK approaches are 77.40% and 76.00%, respectively. This result confirms that the sentence selection performance of the NDTK approach outperforms the original DTK approach. In Fig. 5, we discover that the *R*s of the NDTK's sentence selection are generally higher than the original DTK approach, except for the famous stars. The average *R*s of the sentence selection of NDTK and the original DTK approaches are 76.36% and 74.56%, respectively. For the F-measure *F* aspect, the same tendencies are observed with the *P*s. In Fig. 6, the average *F*s of the sentence selection of NDTK and the original DTK approaches are 0.78 and 0.75, respectively.

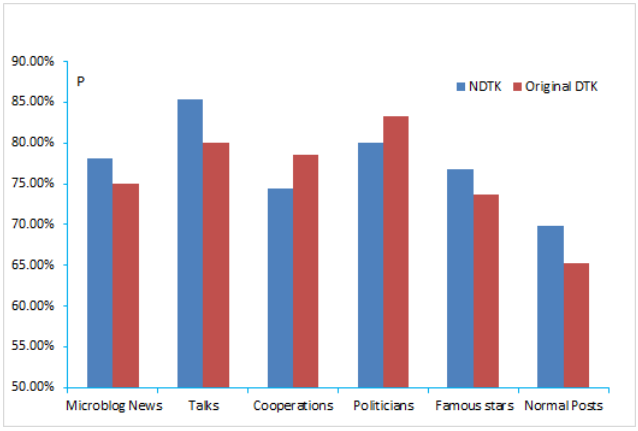


FIGURE 4. Sentence selection precision *P* of the NDTK approach.

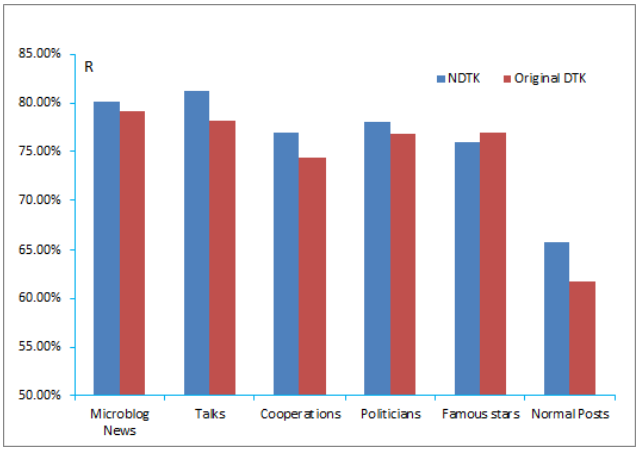


FIGURE 5. Sentence selection recall *R* of the NDTK approach.

Figs. 7∼9 demonstrate the personae social relation extraction performance of the NDTK and original DTK approaches.

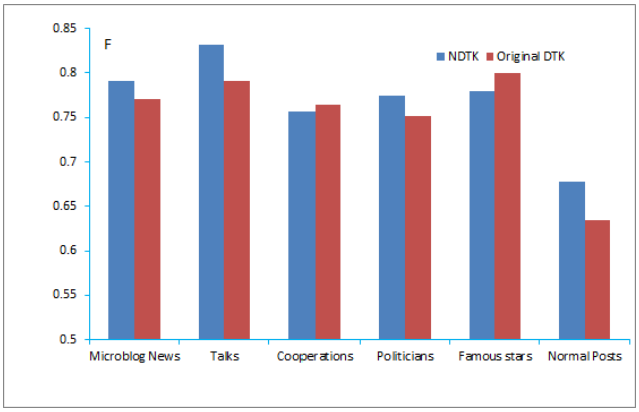


FIGURE 6. Sentence selection F-measure *F* of NDTK approach.

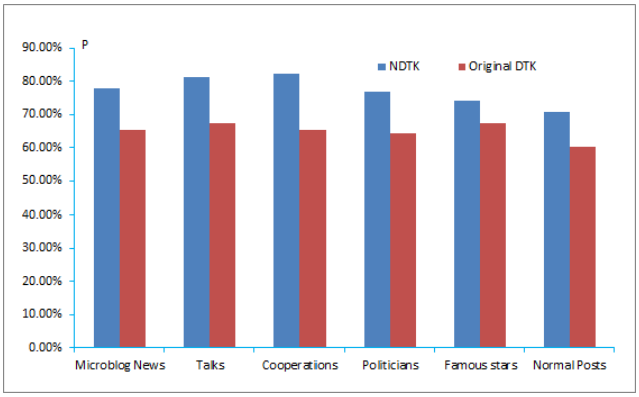


FIGURE 7. Personae social relation extraction precision *P* of NDTK approach.

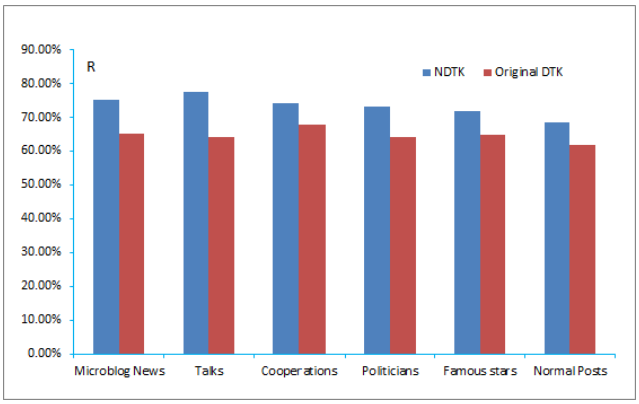


FIGURE 8. Personae social relation extraction recall *R* of NDTK approach.

In Fig. 7, we discover that the *P*s of the NDTK's personae social relation extraction is higher than that of the original DTK approach in microblog news, talks, famous stars, politicians, cooperations, and normal posts. The average *P*s of the personae social relation extraction of the NDTK and the original DTK approaches are 77.23% and 66.90%, respectively. The average *P*s of the personae social relation extraction of the NDTK approach improved by approximately 10% compared with the original DTK approach. This result indicates that the personae social relation extraction of the NDTK approach outperforms the original DTK approach. In Fig. 8,

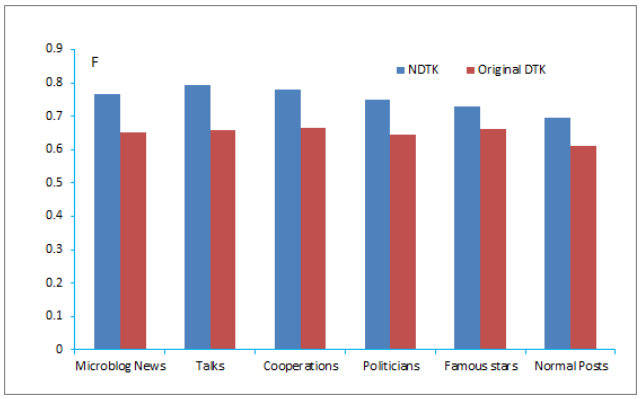


FIGURE 9. Personae social relation extraction F-measure *F* of NDTK approach.

the *R*s of the NDTK's personae social relation extraction are generally higher than those of the original DTK approach. The average *R*s of the personae social relation extraction of the NDTK and DTK approaches are 73.40% and 64.72%, respectively. The average *R*s of the personae social relation extraction of the NDTK approach improved by approximately 9% compared with those of the original DTK approach. For the F-measure *F*s in Fig. 9, the same tendency is observed with the *P*s and *R*s. The average *F*s of the personae social relation extraction of NDTK and the original DTK approaches are 0.75 and 0.64, respectively.

2) EVALUATION OF THE RFWCA

To obtain a more accurate experimental validation, we discard the error posts from 16,532 microblog posts and retain 8,870 microblog posts, which contain 4,534 topic microblog posts (include 5 topics: microblog news, cooperations, talks, politicians, and famous stars), 4,336 normal posts. These relation FWs are classified by our improved *RFWC*. For example, in Fig. 2, the entity words "奥巴马(Obama)" and "默克尔(Merkel)", the relation FWs are "通话(spoke), 商讨(discuss), 希腊(Greek), 债务(Debt), 危机(Crisis)". We considered that "通话" and "商讨" are the correct relation FWs describing the relations between

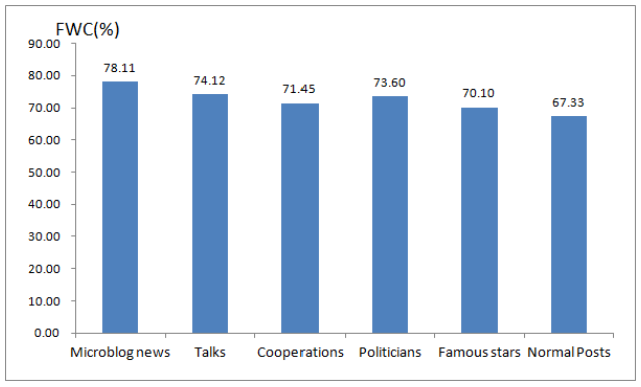


FIGURE 10. FW correct rate of relation feature words.

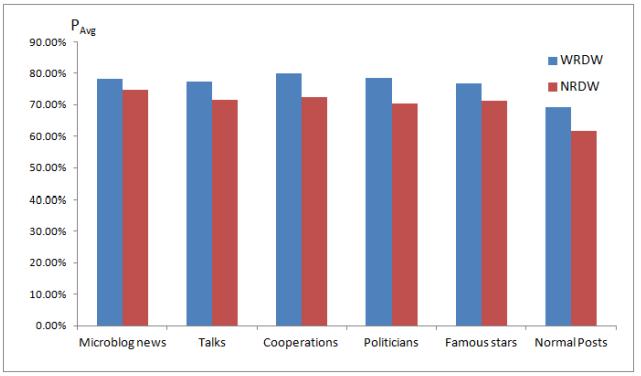


FIGURE 11. Evaluate P_Avg of RFWCA.

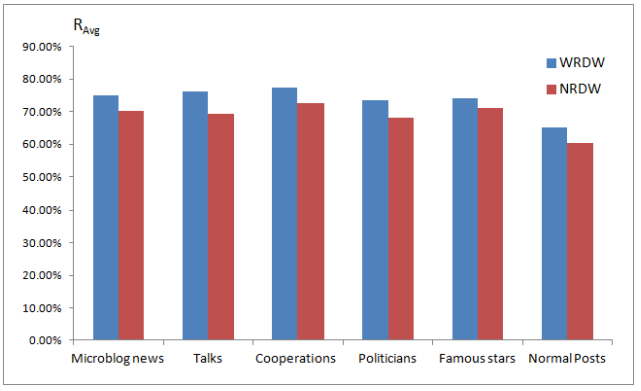


FIGURE 12. Evaluate R_Avg of RFWCA.

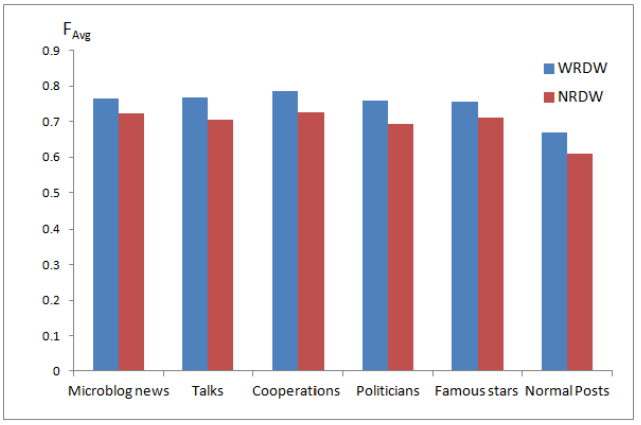


FIGURE 13. Evaluate F_Avg of RFWCA.

"奥巴马" and "默克尔". Eq. (18) can determine the correct or incorrect words.

Fig. 10 shows the *FW* correct rates in the different topics. The *FWC*s of microblog news, talks, cooperations, politicians, famous stars, and normal posts are 78.11%, 74.12%, 71.45%, 73.60%, 70.10%, and 67.33%, respectively. The average *FWC* is 72.41%; the highest *FWC* is 78.11%; and the lowest *FWC* is 67.33%.

After retrieving all relation FWs from the topic microblog posts. We classify these relations into four types: friend, enemy, work and family. To measure the performances

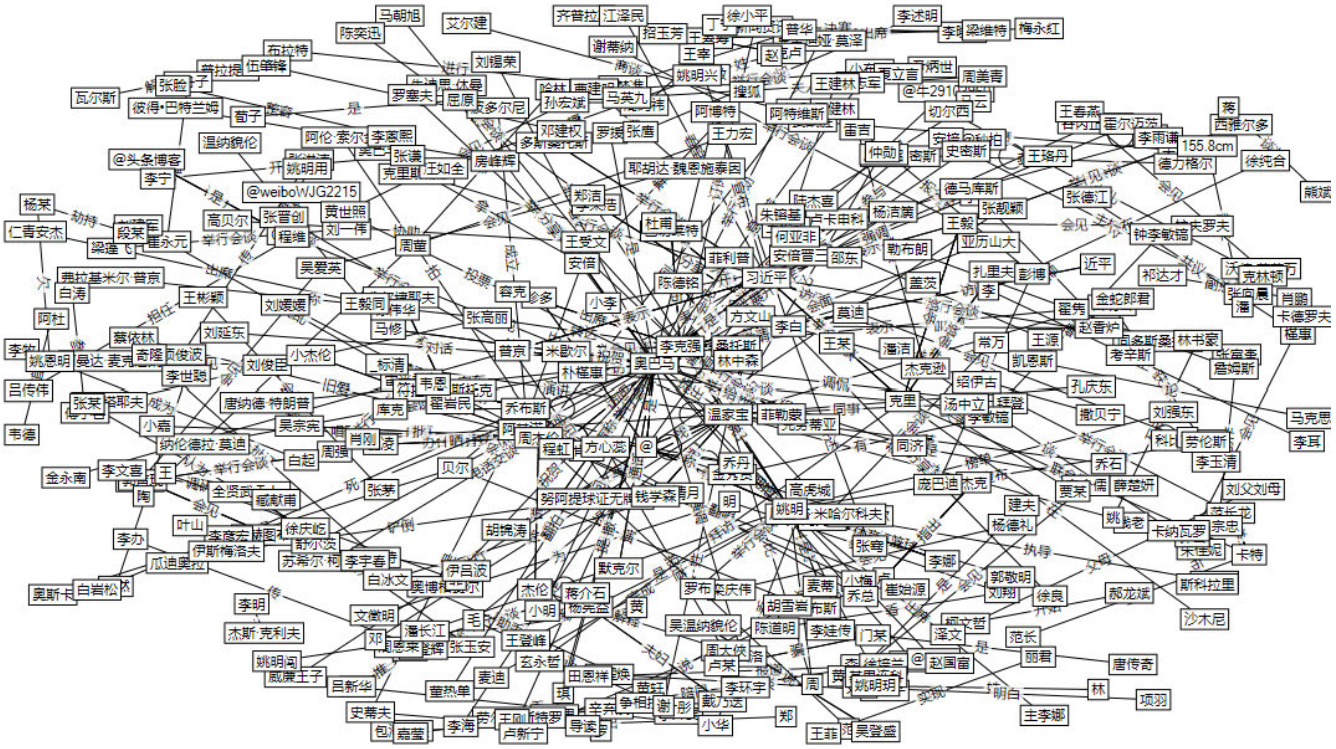


FIGURE 14. Knowledge graph of the personae social relations without RFWCA.

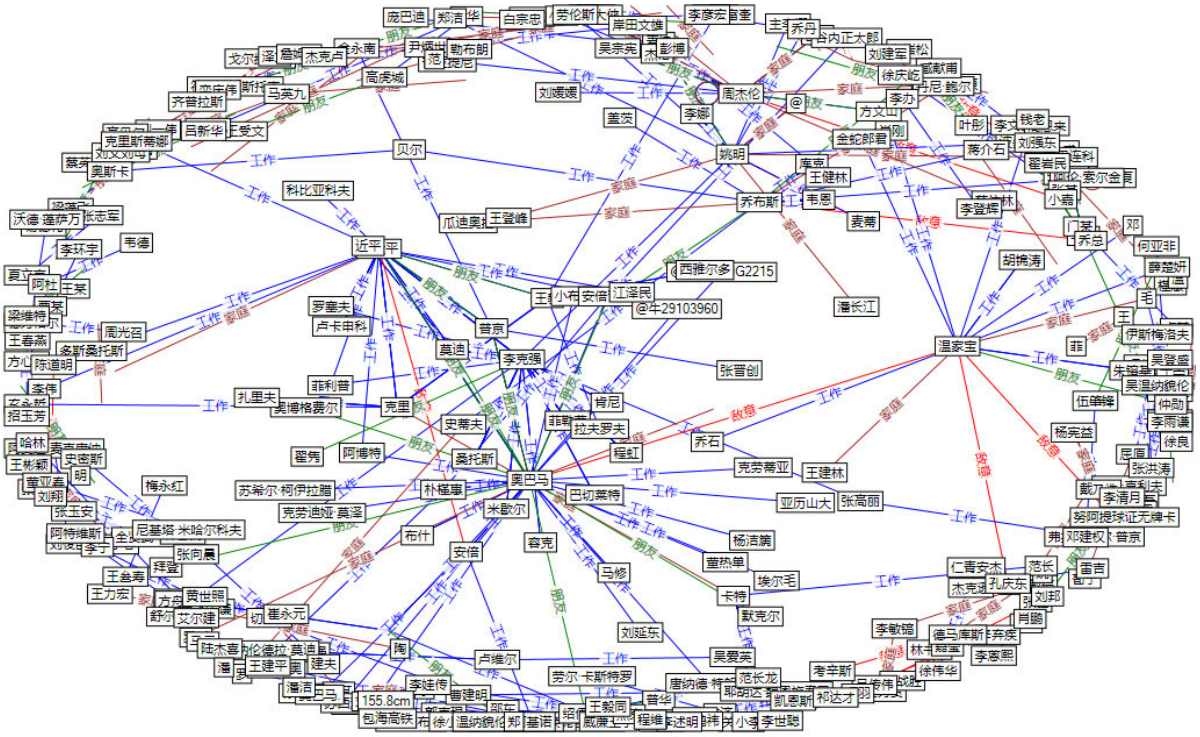


FIGURE 15. Knowledge graph of the personae social relations with RFWCA.

of RFWCA, we build two datasets. The first one (denoted WRDW) processes the dataset with relation feature words. The second (denoted NRDW) processes the dataset without relation feature words. In two processes, WRDW and NRDW are input directly into RFWCA. Then, we retrieved P_Avg, R_Avg and F_Avg. All relation words retrieved from all relation

dependency trigrams are directly divided into the four word types: friend, enemy, work and family. P_Avgs, R_Avgs, and F_Avgs in Figs. 11∼13 of WRDW are higher than those of NRDW. The average P_Avg of WRDW is 76.63%, while that of NRDW is 70.30%. The average P_Avg of WRDW increases by 6.33%. The average R_Avg of WRDW is 73.56%, while that of NRDW is 68.63%. The average R_Avg of WRDW increases by 4.93%. The average F_Avg of WRDW is 0.75, while that of NRDW is 0.69. The average F_Avg of WRDW increases by 0.06.

D. CLASSIFICATION OF RELATION GRAPHS

Using our improved NDTK and proposed RFWCA algorithms, we construct a knowledge graph based on personae social relations for five topics (microblog news, cooperations, talks, politicians, and famous stars) of Chinese microblog posts. We develop the visual interaction interface that we use for node-XL [44], [45] for the knowledge graph. Fig. 14 is a fragment of the social relation graph without RFWCA. Fig. 14 includes the social relations of approximately 600 personae because the personae social relations cover a wide range of topics. There are many relations between two person entities. Parts of these relations are often repetitive. Hence, the structure of the knowledge graph is complex. The different relation types are difficult to distinguish. We cannot mark the relation types in Fig. 14.

We provide a simplified knowledge graph in Fig. 15 by using our improved NDTK and our proposed RFWCA. Fig. 15 includes the same 600 personae, reduces redundant relationships, and makes the structure of the knowledge graph clear. The personae social relations of two entities can be distinguished relatively. In Fig. 15, we assign the different types of relations with the different colors as follows: work, blue; family, brown; friend, green; and enemy, red. Fig. 15 shows clearer entities and relations between entities than Fig.14.

V. CONCLUSION AND FUTURE WORK

In this paper, we take microblog posts as an example to tentatively study the relation extraction of short text. Some conclusions are listed as follows:

- By utilizing the original DTK approach, we propose the NDTK algorithm and seven novel rules for extracting the relation FWs.
- We propose an *FW* words classification algorithm that can classify FWs into different relation types, such as work, family, friend, and enemy.
- Finally, we experimentally evaluate our proposed method to prove the rules, our improved NDTK algorithm and our proposed RFWCA. The experimental results demonstrate good performance for our approaches.

In the future, we will extract personae social relations with microblog posts of more topics and construct their knowledge graph.

ACKNOWLEDGMENT

The authors especially thank Prof. X. He at Zhejiang University for the helpful comments and suggestions.

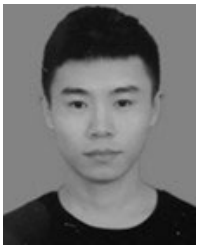
REFERENCES

[1] Y. J. Du and Y. Wu, "Constructing and analysizing the knowledge graph based on micro blog community," *Xihua J. (Nature)*, vol. 89, no. 1, 2015, pp. 27–35.
[2] Accessed: Apr. 7, 2016. [Online]. Available: http://www.google.com/insidesearch/features/search/knowledge.html
[3] Accessed: Apr. 7, 2016. [Online]. Available: http://blogs.bing.com/search/2013/03/21/understand-yourworld-with-bing/
[4] N. Jayaram, A. Khan, C. K. Li, X. Yan, and R. Elmasri, "Querying knowledge graphs by example entity tuples," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2797–2811, Oct. 2015.
[5] M. Arenas, B. C. Grau, E. Kharlamov, Š. Marciuška, and D. Zheleznyakov, "Faceted search over RDF-based knowledge graphs," *J. Web Semantics*, vols. 37–38, pp. 55–74, Mar. 2016.
[6] M. Rospocher, M. van Erp, P. Vossen, A. Fokkens, I. Aldabe, G. Rigau, A. Soroa, T. Ploeger, and T. Bogaard, "Building event-centric knowledge graphs from news," *J. Web Semantics*, vols. 37–38, pp. 132–151, Mar. 2016, doi: 10.1016/j.websem.2015.12.004.
[7] W. Shen, J. Wang, and J. Han, "Entity linking with a knowledge base: Issues, techniques, and solutions," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 443–457, Feb. 2015.
[8] Z. Hai, K. Chang, J. J. Kim, and C. C. Yang, "Identifying features in opinion mining via intrinsic and extrinsic domain relevance," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 623–634, Mar. 2014.
[9] J. J. Jung, "Online named entity recognition method for microblog posts in social networking services: A case study of Twitter," *Expert Syst. Appl.*, vol. 39, no. 9, pp. 8066–8070, 2012.
[10] F. Li, Y. J. Du, H. Y. Zhao, and Z. G. Feng, "Two-phase strategy of Chinese named entity recognition in micro-blog," *J. Comput. Inf. Syst.*, vol. 10, no. 19, pp. 8421–8428, 2014.
[11] *Assessment of Detection and Recognition of Entities and Relations Within and Across Documents*, Autom. Content Extraction 2008 Eval. Plan (ACE08), Linguistic Data Consortium, Univ. Pennsylvania, Philadelphia, PA, USA, 2008.
[12] Accessed: Aug. 5, 2019. [Online]. Available: http://dc.idcquan.com/jfjs/78223.shtml
[13] N. A. Chinchor, "Overview of MUC-7/MET-2," in *Proc. 7th Message Understand. Conf. (MUC-7)*, Fairfax, VA, USA, Apr./May 1998.
[14] X. Zhou, L. P. Liu, X. Luo, H. Chen, L. Qing, and X. He, "Joint entity and relation extraction based on reinforcement learning," *IEEE Access*, vol. 7, pp. 125688–125699, 2019.
[15] S. Brin, "Extracting patterns and relations from the world wide Web," in *The World Wide Web and Databases*. Berlin, Germany: Springer, 1999, pp. 172–183.
[16] Y. Matsuo, J. Mori, M. Hamasaki, T. Nishimura, and H. Takeda, "POLYPHONET: An advanced social network extraction system from the Web," *Web Semantics Sci. Services Agents World Wide Web*, vol. 5, no. 4, pp. 262–278, 2007.
[17] T. Nie, D. Shen, Y. Kou, G. Yu, and D. Yue, "An entity relation extraction model based on semantic pattern matching," in *Proc. 8th Web Inf. Syst. Appl. Conf.*, Oct. 2011, pp. 7–12.
[18] Z. Xu, X. Luo, S. Zhang, X. Wei, L. Mei, and C. Hu, "Mining temporal explicit and implicit semantic relations between entities using Web search engines," *Future Generat. Comput. Syst.*, vol. 37, pp. 468–477, Jul. 2014.
[19] C. Zhang, Y. Zhang, W. Xu, Z. Ma, Y. Leng, and J. Guo, "Mining activation force defined dependency patterns for relation extraction," *Knowl.-Based Syst.*, vol. 86, pp. 278–287, Sep. 2015.
[20] N. Kambhatla, "Combining lexical, syntactic and semantic features with maximum entropy models for extracting relations," in *Proc. ACL Interact. Poster Demonstration Sessions*, Jul. 2004, pp. 178–181.
[21] W. Che, T. Liu, and S. Li, "Automatic entity relation extraction," *J. Chin. Inf. Process.*, vol. 19, no. 2, pp. 1–6, 2005.
[22] S. Xia and D. Lehong, "Feature-based approach to Chinese term relation extraction," in *Proc. Int. Conf. Signal Process. Syst.*, May 2009, pp. 410–414.
[23] H. Liu, C. Jiang, C. Hu, and L. Zhang, "Efficient relation extraction method based on spatial feature using ELM," *Neural Comput. Appl.*, vol. 27, pp. 271–281, Feb. 2016.

[24] W. C. Huang, S. S. Fan, L. Y. Xiong, and M. S. Zhong, "People relation extraction method based on feature selection," *Sci. Technol. Eng.*, vol. 15, no. 3, pp. 254–259, 2015.
[25] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *J. Mach. Learn. Res.*, vol. 3, pp. 1083–1106, Feb. 2003.
[26] H. H. Yu, L. H. Qian, G. D. Zhou, and Q. M. Zhu, "Tree kernel-Chinese semantic relation extraction based on unified syntactic and entity semantic tree," *J. Chin. Inf. Process.*, vol. 24, no. 5, pp. 17–23, 2010.
[27] G. D. Zhou, M. Zhang, D. Hong, and J. Q. M. Zhu, "Tree kernel-based relation extraction with context-sensitive structured parse tree information," in *Proc. EMNLP*, Jan. 2007, pp. 728–736.
[28] G. D. Zhou, L. H. Qian, and J. X. Fan, "Tree kernel-based semantic relation extraction with rich syntactic and semantic information," *Inf. Sci.*, vol. 180, no. 8, pp. 1313–1325, 2010.
[29] H. W. Chun, C. H. Jeong, S. K. Song, Y. S. Choi, S. P. Choi, and W. K. Sung, "Relation extraction based on composite kernel combining pattern similarity of predicate-argument structure," *Commun. Comput. Inf. Sci.*, vol. 264, no. 4, pp. 269–273, 2011.
[30] L. Li, J. Zhang, L. Jin, R. Guo, and D. Huang, "A distributed meta-learning system for Chinese entity relation extraction," *Neurocomputing*, vol. 149, pp. 1135–1142, Feb. 2015.
[31] M. Choi and H. Kim, "Social relation extraction from texts using a support-vector-machine-based dependency trigram kernel," *Inf. Process. Manage.*, vol. 49, no. 1, pp. 303–311, 2013.
[32] C. H. Huang, J. Yan, and F. Hou, "A text similarity measurement combining word semantic information with TF-IDF method," *J. Comput.*, vol. 34, no. 5, pp. 854–856, 2011.
[33] Y. J. Du and Y. F. Hai, "Semantic ranking of Web pages based on formal concept analysis," *J. Syst. Softw.*, vol. 86, no. 1, pp. 187–197, 2013.
[34] Y. J. Du, Y. F. Hai, and C. Z. Xie, "An approach for selecting seed urls of focused crawler based on user-interest ontology," *Appl. Soft Comput.*, vol. 14, no. C, pp. 663–676, 2014.
[35] W. J. Liu and Y. J. Du, "A novel focused crawler based on cell-like membrane computing optimization algorithm," *Neurocomputing*, vol. 123, pp. 266–280, Jan. 2014.
[36] Q. Liu and S. J. Li, "Word similarity computing based on how-net," *Comput. Linguistics*, vol. 7, no. 2, pp. 59–76, 2002.
[37] Y. J. Du, Q. Q. Pen, and Z. Q. Gao, "A topic-specific crawling strategy based on semantic similarity," *Data Knowl. Eng.*, vol. 88, pp. 75–93, Nov. 2013.
[38] Y. J. Du, C. X. Li, Q. Hu, X. L. Li, and X. L. Chen, "Ranking Webpages using a path trust knowledge graph," *Neurocomputing*, vol. 269, pp. 58–72, Dec. 2017.
[39] C. H. Wang, M. Zhang, S. P. Ma, and L. Y. Ru, "Automatic hot event detection using both media and user attention," *J. Comput. Inf. Syst.*, vol. 4, no. 3, pp. 985–992, 2008.
[40] S. L. Wu, C. H. Song, H. D. Chen, and S. Q. He, "The soccer semantic event detection using multi-attribute and decision tree," *J. Comput. Inf. Syst.*, vol. 6, no. 2, pp. 585–592, 2010.
[41] J. Xu, J. H. Gan, X. M. Yao, and L. M. Zhang, "Concept meaning acquisition based on HowNet and its application in the construction of taxonomy," *Int. J. Performability Eng.*, vol. 14, no. 7, pp. 1459–1467, 2018.
[42] Iflytek Company, Ltd. *Harbin Institute of Technology*. Accessed: 2019. [Online]. Available: http:www.ltpcloud.com
[43] M. Melucci and A. Paggiaro, "Evaluation of information retrieval systems using structural equation modeling," *Comput. Sci. Rev.*, vol. 31, pp. 1–18, Feb. 2019.
[44] *NodeXL: Network Overview, Discovery and Exploration for Excel*. Accessed: 2018. [Online]. Available: http://nodexl.codeplex.coms
[45] J. H. Yan, C. Y. Wang, W. L. Cheng, M. Gao, and A. Y. Zhou, "A retrospective of knowledge graphs," *Frontiers Comput. Sci.*, vol. 12, no. 1, pp. 55–74, 2018.

YAJUN DU received the Ph.D. degree in traffic information engine and control from the School of Computer and Communicate, Southwest Jiaotong University, in 2005. He is currently a Professor in computer science with Xihua University. He has published several articles in information retrieve, search engine, focused crawler, and knowledge graph. He is serving in committees of Chinese information and PC member for several leading international/conferences WMSE, ICIC, CCIR, CCKS, and SMP. His experiences and researchs works focus on information retrieval, software engineering, search engine, web mining, and computer networks.

FANGHONG SU received the bachelor's degree in computer science and technology from the School of Computer and Soft Engineering, Xihua University, in 2002. He is currently working with Sichuan Lewei Technology Company, Ltd. His experience and research work focuses on knowledge graph.



ANZHENG YANG received the M.S. degree in computer science and technology from the School of Computer and Soft Engineering, Xihua University, in 2016. He has published two articles in information retrieval, search engine, focused crawler, and knowledge graph. His experience and research work focuses on social networks and software engineering.

XIANYONG LI received the D.S. degree in computer science from Chongqing University, in 2014. He is currently a Lecturer of computer science with Xihua University. He has published more than ten academic articles in journals. His research interests include information retrieval, web mining, fault-tolerant computing, interconnection networks, and graph theory.

YONGQUAN FAN received the D.S. degree in traffic information engine and control from SWJTU, in 2010. He is currently an Assistant Professor in software engineering with Xihua University. He has published several articles. His experiences and researchs works focus on information retrieval, information filter, search engine, and web mining.