SPIRIT: A Tree Kernel-based Method for Topic Person Interaction Detection

Yung-Chun Chang, Chien Chin Chen, and Wen-Lian Hsu, Fellow, IEEE

Abstract—The development of a topic in a set of topic documents is constituted by a series of person interactions at a specific time and place. Knowing the interactions of the persons mentioned in these documents is helpful for readers to better comprehend the documents. In this paper, we propose a topic person interaction detection method called SPIRIT, which classifies the text segments in a set of topic documents that convey person interactions. We design the rich interactive tree structure to represent syntactic, context, and semantic information of text, and this structure is incorporated into a tree-based convolution kernel to identify interactive segments. Experiment results based on real world topics demonstrate that the proposed rich interactive tree structure effectively detects the topic person interactions and that our method outperforms many well-known relation extraction and protein-protein interaction methods.

Index Terms—Text mining, natural language processing, classification

1 INTRODUCTION

The web has become a powerful medium for disseminating information on a diverse range of topics, such as political issues and financial events. While people can easily find documents that cover various perspectives of a topic, they often have difficulty assimilating the information in large documents. This information overload problem has motivated the development of topic mining methods to help readers navigate the seas of topic information. For instance, Nallapati et al. [1] and Feng and Allan [2] grouped topic documents into clusters that each represents a theme in a topic and that are then connected chronologically to form a timeline of the topic. Recently, Chen and Chen [3] summarized the incidents of a topic timeline to help readers understand the story of a topic more quickly. Although the extracted themes and summaries distilled the topic contents clearly, readers still needed to spend a considerable amount of time to comprehend the extracted information on unfamiliar topics.

Generally speaking, a topic is associated with specific times, places, and persons [1]. The interactions between the persons (called topic persons hereafter) can be said to constitute the storyline of the topic, and thus knowing the topic person interactions is critical for readers to comprehend several related topic documents. In this paper, we investigate the detection of topic person interactions, which involves identifying text segments in topic documents that convey interactions between topic persons. For instance, given the documents about the 2012 U.S. presidential election, the detection of topic person interaction designates the text segments that mention the interactions of the key persons of different camps, as shown in Fig.1. In addition to facilitating topic comprehension, topic person interaction detection can also benefit many information retrieval and natural language processing tasks. Topic mining methods, for example, can incorporate the segments of person interactions into the generated topic summaries to present the development of topics. Moreover, as the expressions of person interactions are usually combined with sentiments, sentiment analysis methods which detect textual units with positive or negative orientations can isolate the segments of person interactions to further filter their results. On the other hand, question answering systems can provide correct answers to who-type questions regarding person interactions. Hence, detecting the interactions of topic persons is worth investigating.

Detecting topic person interactions is a new research area. To the best of our knowledge, only Chang et al. [4], [5] have investigated this research subject in the past. Specifically, the authors considered interaction detection as a classification problem and developed a feature-based detection method named FISER to classify text segments mentioning interactions between topic persons. Although this method explores various text features for the classification task, it ignores the syntactic structure of the text, which is useful for discovering the relations between named entities [6]. A number of bioinformatic studies (e.g., [7], [8]) have looked into the problem of protein-protein interaction (PPI) detection, which identifies text segments from medical documents that refer to interactions between proteins. Similarly, in the field of relation extraction (RE), an important task is to discover text segments mentioning the relations between entities (e.g., persons) [6]. However, PPI detection aims to discover permanent interactions between proteins, such as covalent and non-covalent bindings, while the relations studied in RE (e.g., employee-of) are predefined and static. By contrast, person interactions exemplify types of

• Yung-Chun Chang is with the Department of Information Management, National Taiwan University, Taipei, Taiwan, Republic of China. E-mail: d89725007@ntu.edu.tw
• Chien Chin Chen is with the Department of Information Management, National Taiwan University, Taipei, Taiwan, Republic of China. E-mail: patonchen@ntu.edu.tw (Corresponding author)
• Wen-Lian Hsu is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan, Republic of China. E-mail: hsu@iis.sinica.edu.tw
human behaviors (e.g., compliment, criticism, collaboration, and competition) that make people consider or influence each other [9]. The interactions between persons are therefore diverse and changeable, which makes their detection intractable. Taking Fig. 1 as an example, the Democratic Party candidate, incumbent President Barack Obama, generally criticized Mitt Romney (the Republican candidate) for his political views during the campaign. However, after President Obama won the election, he broke bread with Mitt Romney at the White House, and even considered offering him a position in the new cabinet. Furthermore, references to person interactions may be located across different clauses, and consist of several verbs and named entities that prevent their detection. As a result, topic person interaction detection is a difficult and challenging task.

In this work, we present a novel topic person interaction detection method called SPIRIT (Scouting Person Interaction using Rich Interactive Tree), which detects the text segments that mention the interactions between topic persons. We focus specifically on Chinese interaction detection because the growth in the number of Chinese documents has been so enormous that Chinese will soon become the second largest language on the web. As there are valuable sources of information in Chinese, many research groups and projects (e.g., NTCIR and SIGHAN) have been established to promote widespread interest in all aspects of Chinese language processing. When given a set of topic documents, SPIRIT first decomposes them into text segments, each of which contains topic persons. Next, the text segments are classified into six categories in accordance with their syntactic characteristics and are represented by a rich interactive tree (RIT) structure, which is the shortest path-enclosed tree of a segment’s parse tree integrated with the segment’s context and semantic information. Finally, the RITs of the text segments together with the convolution tree kernel [10] are applied to the support vector machine (SVM) [11] to classify interactive segments. Results of experiments based on real-world datasets demonstrate that the proposed RIT structure successfully exploits the syntactic structures, interaction semantics, and segment context relevant to person interactions. Consequently, SPIRIT outperforms many well-known relation extraction and PPI methods.

The remainder of this paper is organized as follows. In the next section, we review related works. Section 3 introduces the structure of SPIRIT, and its system performance is evaluated in Section 4. Finally, we present our conclusions in Section 5.

2 RELATED WORK

In this section, we review the selection techniques used in relation extraction. Then, we consider a number of protein-protein interaction detection methods and explain how these approaches differ from topic person interaction detection.

2.1 Relation Extraction

Our research is closely related to relation extraction (RE), in which the goal is to discover relations between entities mentioned in the text. RE was first introduced as a part of the template element task in the sixth Message Understanding Conference (MUC6) and was soon formulated as a regular task in MUC7. Now, it has become an important research issue and is intensively promoted by the Message Understanding Conferences (MUCs) and the Automatic Content Extraction (ACE) program. As defined by the ACE program, an RE’s entity is an object or a set of objects in the world, and their relations can be expressed explicitly or implicitly in a given text. For instance, the sentence “Bill Gates is the chair and chief software architect of Microsoft Corporation” implicitly states that Bill Gates (ACE entity type: PERSON.Name) is an employee of (ACE relation type: EMPLOYMENT.exec) Microsoft Corporation (ORGANIZATION.Commercial). The results of RE can benefit...
a variety of applications related to information retrieval and natural language processing, such as question answering. For instance, given a query like "What company does Bill Gates work for?", an RE-enhanced search engine would answer Microsoft instead of returning pages about Microsoft’s products. Due to its practicality, the ACE program has collected a series of relation extraction corpora for benchmarking, which examine 23 types of relations among persons, organizations, locations, facilities, and geo-political entities.

Previous works generally consider relation extraction as a supervised classification task. Given a set of training segments (e.g., sentences) regarding a specific entity relation, a supervised classification algorithm is employed to learn a relation classifier. The classifier then determines (classifies) whether a new text segment expresses the relation or not. In order to employ a classification algorithm, features are extracted from text segments. Depending on the type of features used, RE methods can be classified as either the feature-based or the tree kernel-based approach [12]. The feature-based methods exploit training segments to identify representative text features for relation extraction. For example, Zhou et al. [13] employed SVM to classify relation segments. They extracted various lexical, syntactic, and semantic features from text segments, and conducted exhaustive experiments to examine their effects. Their experiment results showed that the phrase chunking of texts is effective for relation extraction. On the other hand, Jiang and Zhai [14] systematically explored the syntactic parse tree and dependency parse tree of a text for relation extraction. The authors demonstrated that simple text features (e.g., bag-of-word and non-conjunctive entity attribute features) are sufficient to achieve superior relation extraction performance, whereas over-inclusion of complex features (e.g., adding grammar productions into the syntactic parse tree) actually impairs the performance.

Selecting representative features is challenging and generally requires extensive feature engineering. Although the parse tree of a text has been exploited by several feature-based methods (e.g. [13]), the selected features have difficulty taking into account the text's syntactic structure, which affects the relation extraction performance [15]. This problem has been addressed with the development of many tree kernel-based methods which rely on a tree kernel that can implicitly explore all sub-structures of a parse tree in a high dimensional feature space and calculate the similarity of parse trees in an efficient manner. Moschitti [16] first applied the convolution tree kernel (CTK) [10] to the problem of semantic role labeling, the task of determining the semantic relationship between a predicate and arguments mentioned in a text segment, which can be considered as the predecessor of relation extraction. The convolution tree kernel examines the syntactic parse tree of a segment, and then assigns it a semantic relationship if it is syntactically similar to the segments of that relationship in a training corpus. Instead of examining the entire parse tree, Zhang et al. [6] incorporated the shortest path-enclosed tree (SPT), which is the sub-parse tree enclosed by the shortest path linking two entities into the CTK for relation extraction. Their experiments demonstrate that the SPT clearly expresses the syntactic relation between entities, and achieves a superior relation extraction performance on many ACE corpora. As a result of their great success, successive RE methods began to combine the CTK with SPT.

### 2.2 Protein-protein Interaction Detection

Our research is also related to protein-protein interaction (PPI) detection [17] that focuses on recognizing the protein interactions mentioned in the biomedical literatures. In this field of research, determining protein interaction partners is crucial for understanding both the functional role of individual proteins and the organization of the entire biological process. Similar to RE, a great portion of PPI detection methods are feature-based. These methods extract lexical, syntactic, and semantic features from a text to construct classification models which distinguish text segments that specify protein interactions. For instance, Ono et al. [17] manually defined a set of syntactic rule-based features covering words and part-of-speech patterns. The authors also developed content-matching rules which examine interaction keywords in order to recognize the protein-protein interaction described in a sentence. Nevertheless, these features have difficulty representing the structured and syntactic dependency of text, which are essential for protein-protein interaction detection. In light of this issue, many tree kernel-based PPI detection methods have been devised. An example of this is Qian and Zhou’s [7] development of a novel tree kernel-based PPI detection method in which the parse tree of a text generated by a constituent syntactic parser is revised by the shortest dependency path between two proteins. The authors showed that the revised parse tree is effective in discriminating the PPIs within texts. More comprehensive reviews in PPI and RE can be found in [5].

Our research differs from RE and PPI detection as they aim to determine the static and permanent relations between entities, whereas our research studies person interactions which are diverse and changeable. To capture the sophisticated nature of person interactions, we integrated the content, syntactic, and semantic information of a text into a rich interactive tree structure that is used by SPIRIT to discriminate interactive text segments. The proposed RIT structure depicts the syntactic structure of a text, which has been shown to be effective in discovering the relations between named entities. In addition, the verbs (i.e., content) in the RIT are examined as interactions because they are generally expressed by verbs. The verbs’ concepts (i.e., semantics) and their concept similarity are explored to decorate the RIT with interactive semantics to sense diverse person interactions.

### 3 SPIRIT System

In this section, we present SPIRIT, which automatically detects text segments (called interactive segments hereafter) that convey person interactions in a set of Chinese topic
documents. Fig. 2 displays the system architecture of SPIRIT, which is comprised of four key components: candidate segment generation, segment structure generalization, rich interactive tree construction, and convolution tree kernel classification. The candidate segment generation first decomposes the topic documents into a set of candidate segments, each of which is likely to mention interactions of topic persons in the segments. As a text’s syntactic information is useful in resolving the relationship between entities [6], [13], [18], we invented the Rich Interactive Tree (RIT) structure that depicts the syntactic path of topic persons in a candidate segment’s parse tree. Meanwhile, the segment’s content is examined to ornament the rich interactive tree with interactive semantics. We adopted the convolution tree kernel [10] to measure the similarity between text segments in terms of their RITs. The tree kernel is incorporated into SVM in order to learn a classifier for each structural type to detect (classify) interactive segments in the topic documents. We discuss each component in detail in the following sub-sections.

### 3.1 Candidate Segment Generation

Fig. 3 shows our candidate segment generation algorithm. Before running the algorithm, a Chinese named entity recognizer is employed to label the tokens that represent a person’s name in the topic documents. Let \( P = \{p_1, \ldots, p_l\} \) denote the set of labeled topic persons. For each topic document \( d \), the algorithm examines every person pair \((p, p)\) in \( P \) and extracts text segments from \( d \) that are likely to mention their interactions. As the interaction between \( p \) and \( p \) may be narrated in a sequence of clauses, we hereby consider two types of candidate segments, namely intra-clausal candidate segments and inter-clausal candidate segments. For instance, as shown in Fig. 1, the text segment “論壇後歐巴馬微幅領先羅姆尼 (Obama slightly leading Romney after the debate.)” is an intra-clausal candidate segment, and the segment “由於歐巴馬在上一場總統辯論中表現過於保守，如今共和黨候選人羅姆尼民調呈現領先趨勢” serves as a complement to describe the interaction of Republican presidential candidate Romney show him in the lead.” is an inter-clausal candidate segment. The algorithm first decomposes \( d \) into a sequence of clauses \( C = \{c_1, \ldots, c_m\} \). Next, it processes the clauses in \( C \) one by one, and considers a clause as the initial clause of a candidate segment if it contains topic person \( p \) or \( p \). The algorithm then examines the initial clause and subsequent clauses until it reaches an end clause that contains the topic person \( p \) or \( p \). If the initial clause is identical to the end clause, the algorithm generates an intra-clausal candidate segment. Otherwise, it generates an inter-clausal candidate segment. Note that in Chinese, a period indicates the end of a discourse. If there is a period within the inter-clausal candidate segment, we drop the segment because \( p \) and \( p \) belong to different discourses. In addition, if \( p \) or \( p \) appears more than once in an inter-clausal candidate segment, we truncate all the clauses before the last \( p \) or \( p \) to make the candidate segment concise. By running all topic person pairs of \( P \) over the topic documents, we obtain a candidate segment set \( CS = \{c_1, \ldots, c_s\} \).
Obama and Hu Jintao, but it occurs after the two topic persons in the segment. Moreover, Obama and Hu Jintao appear in different clauses, and the text between them contains more than one verb and different named entities that complicate the interaction detection of the segment.

To reduce the complexity of text structures, we generalize candidate segments into six structural types according to the relative position of the verbs to $p_i$ and $p_j$, which are the topic persons used to generate a candidate segment.

- If all of the verbs of an intra-clausal (inter-clausal) candidate segment occur between $p_i$ and $p_j$, the segment is categorized as intra-inner (inter-inner) candidate segments.
- If the verbs only occur after the last topic person, we categorize it as an intra-outer (inter-outer) candidate segment.
- Finally, if the verbs occur not only between the topic persons but also after the last topic person, we name it as an intra-surrounding (inter-surrounding) candidate segment.

Fig. 5 shows some examples of the six structural types of candidate segments. Note that if a candidate segment does not belong to any structural type (i.e., it is a short segment without any verb), it will be judged as a non-interactive segment. In the experiment section, we examine the effect of the structure generalization on interaction detection.

### 3.3 Rich Interactive Tree Construction

Next, we represent candidate segments with the rich interactive tree structure, which is a segment’s shortest path-enclosed tree (SPT) [6] enhanced by three operators: branching, pruning, and ornamenting. Specifically, the SPT of a candidate segment is the smallest sub-tree of the segment’s syntactic parse tree that links topic persons $p_i$ and $p_j$.

In [6], the authors show that SPT is effective in identifying the relation between two entities mentioned in a text segment. To facilitate comprehension, we exemplify the operators on the example segment that expresses the interaction between 歐巴馬 (Barack Obama) and 胡錦濤 (Hu Jintao). Fig. 6 shows the SPT of the example segment. The three operators utilized by RIT to polish SPT are described in detail as follows.

**3.3.1 RIT Branching:**

In many cases, the information in SPT is insufficient for person interaction detection. In Fig. 4, for example, the term "會談 (engage in colloquy)" and the corresponding syntactic constituent are vital for recognizing the interaction between Obama and Hu Jintao. However, they are excluded from the segment’s SPT as shown in Fig. 6. To include a useful segment context, the branching operator first examines
whether there is a verb behind the last topic person. Moreover, if the verb and the topic person form a verb phrase in the segment’s syntactic parse tree, the verb is treated as a modifier of the topic person and is concatenated into the RIT. As shown in Fig. 7, the branched RIT includes richer context information than the original RIT.

3.3.2 RIT Pruning:

Before we start to train a classifier for interaction detection, the structure of the RIT should be condensed first. This is because SVM is a vector space classification model that hypothesizes that data of different classes form distinct contiguous regions in a high-dimensional vector space [21]. This hypothesis, however, is invalid if data representation is chosen improperly. We noticed that the RITs would contain redundant elements that would influence the classification performance of interaction detection. The pruning operator condenses the RITs via the following procedures.

- **Middle clause removal**: Middle clauses of inter-clause candidate segments may be irrelevant to person interactions. To discriminate middle clauses associated with the topic persons, we adopted the Stanford parser [22] which labels dependencies between text tokens (words). A labeled dependency is a triplet of dependency name, governor token, and dependent token. The labeled dependencies form a directed graph $G = (V, E)$, where each vertex in $V$ is a token and the edges in $E$ denote the set of dependencies. Fig. 8 displays the dependency graph generated from the example segment.

Next, we search for the person dependency path which we defined as the shortest connecting path of the topic persons in $G$. The person dependency path of the example segment is highlighted in red in Fig. 8 and is 東帝汶 arrive 會談 engage in colloquy 與 with 胡錦濤 Hu Jintao. The pruning operator removes a middle clause and all of its elements in the RIT if the clause is not involved in the person dependency path. The clause is pruned because it is unable to associate the topic persons with each other. In Fig. 9, the middle clause "拒絕媒體採訪(refuses to give any interview to the press)" is pruned because it only associates Obama with the press.

- **Stop word removal**: Frequent words are generally too common to discriminate person interactions. Here, we used the well-known Chinese stop word list compiled...
by collecting the most frequent words in the Sinica corpus. The pruning operator removes a word and the corresponding elements in the RIT if it is a stop word. For instance, in Fig. 9, the word “便 (then)” and the corresponding RIT branch are removed from the segment’s RIT because it is a common Chinese word.

- **Grammatical standardization of the tree:** Nodes in an RIT are often duplicated. A node is duplicated if it has a single child and its tag is identical to that of its parent, such as the node VP in the last branch of Fig. 7. Duplicate nodes are misspelled grammatical relations. Since the tree kernel we adopted to compute the similarity of text segments is based on the overlap of RITs, duplicate nodes would degrade our system performance. Therefore, to reduce their influence, the pruning operator simplifies an RIT by deleting all duplicate nodes. As shown in Fig. 9, the pruned RIT is more clear and precise than the original RIT.

![Figure 9](http://www.aclclp.org.tw/use_wlawf.php)

**3.3.3 RIT Ornamenting:**

In addition to syntactical structures, the RIT further explores the semantics of candidate segments. The designed ornament operator examines the verbs of a candidate segment to decorate the RIT with interactive semantics. We first employ the log likelihood ratio (LLR) [21], an effective feature selection method, to learn a set of verbs with interactive semantics. Given a training dataset comprised of interactive and non-interactive segments, LLR employs the following equation to calculate the likelihood that the occurrence of a verb \( v \) in the interactive segments is not random:

\[
LLR_{\text{interactive}}(v) = -2 \log \left[ \frac{p(v | IS)}{p(v | \neg IS)} \right] \times \frac{p(v | IS)(1 - p(v | IS))}{p(v | \neg IS)(1 - p(v | \neg IS))} \times \frac{N(v | IS)}{N(v | \neg IS)},
\]

where IS denotes the set of interactive segments in the training dataset; \( N(IS) \) and \( N(\neg IS) \) are the numbers of interactive and non-interactive segments, respectively; and \( N(v | IS) \) is the number of interactive segments having \( v \). The probabilities \( p(v), p(v | IS), \) and \( p(v | \neg IS) \) are estimated by using maximum likelihood estimation. A verb with a large LLR value is closely associated with the interactive segments. We rank the verbs in the training dataset based on their LLR values and select the top 150 to compile an interactive verb list.

Candidate segments, especially those that are inter-clausal, generally contain several verbs, although many of these are irrelevant to topic persons and would distract the detection of person interactions. Again, we utilize the person dependency path to dig out verbs associated with the topic persons. As illustrated in Fig. 8, the path excludes the verb 拒絕 (refuse), which expresses the action of Obama to the press and is irrelevant to the interaction of the topic persons. A simple approach to examine the interactive semantics of a person dependency path is to match the verbs in the path with the interactive verb list. The word-matching approach, however, overlooks many interactive segments because it omits synonyms of interactive verbs. To remedy this problem, the ornament operator further considers the concepts of verbs and measures verb similarity by means of a concept taxonomy. We utilize E-HowNet [23], a well-known Chinese lexical taxonomy compiled by linguistic experts, to acquire the concepts of a verb. Similar to WordNet [24], words in E-HowNet are grouped into concepts that are interlinked to form a taxonomy in terms of conceptual semantics or relations. Furthermore, useful statistics such as word and concept frequencies based on a huge text corpus are provided, making E-HowNet an important knowledge base for Chinese NLP research [25], [26]. We employed the following equation to measure the similarity of two concepts in E-HowNet:

\[
sim(c_i, c_j) = \max_{c \in \text{supp}(c_i, c_j)} \left[-\log p(c)\right],
\]

where \( \text{supp}(c_i, c_j) \) is the set of super concepts that subsume concepts \( c_i \) and \( c_j \), and \( p(c) \) is the probability of concept \( c \) derived by the following maximum likelihood estimation:

\[
p(c) = \frac{\text{freq}(c)}{N},
\]

where \( \text{freq}(c) \) is the frequency of all words subsumed by concept \( c \) and \( N \) is the total word frequency in E-HowNet. The negative log likelihood of Equation 2 measures the information content [27] of a concept. A specific concept would have a high information content because its probability \( p(c) \) is low. Based on this, \( c \) and \( c_j \) are similar if they are subsumed by a specific concept. Next, the following equation considering word polysemy is used to determine the similarity of a verb \( v \) in the person dependency path and a verb \( v' \) in the interactive verb list:

\[
sim(v, v') = \max_{c \in \text{dep}(v') \cap \text{dep}(v')} \left[ \frac{\sim(c_i, c_j)}{\log N} \right],
\]

where \( \text{dep}(v) \) denotes the set of concepts \( v \) belongs. The denominator is a normalization factor that makes the similarity range from 0 to 1.

On the whole, the ornament operator processes the verbs in the person dependency path one by one. If a verb matches an interactive verb or its similarity to an interactive verb is larger than a predefined threshold \( \zeta \), we incorporate the RIT with interactive semantics by adding an interactive verb tag IV as a child of the root. Fig. 10 shows the ornamented result of the example segment in which an IV
tag shown in gray is appended to the root of the RIT because the verb 会話 (engage in colloquy) is highly associated with the verb 深説 (converse) in our interactive verb list.

Fig. 10. The ornamenting operation of the rich interactive tree of the example segment.

### 3.4 Convolution Tree Kernel Classification

For each structural type that we defined in Section 3.2, we prepare a training dataset \( D = \{ \langle \text{RIT}_1, y_1 \rangle, \ldots, \langle \text{RIT}_n, y_n \rangle \} \), where \( \text{RIT} \) is the rich interactive tree of the training segment and \( y \) is the segment’s class label such that \( y = 1 \) if the segment is an interactive segment. Otherwise, \( y = -1 \). The training of the SVM is to find \( \alpha \), ... \( \alpha \), such that

\[
\max \left\{ \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_i \alpha_j y_i y_j \phi(\text{RIT}_i) \cdot \phi(\text{RIT}_j) \right\} \tag{5}
\]

subject to \( \sum_{i=1}^{M} \alpha_i y_i = 0, \alpha \geq 0 \) for all \( 1 \leq i \leq M \),

where \( \alpha \) is the Lagrange multiplier associated with the \( i \)-th training segment and \( \phi(\text{RIT}_i) \) is the feature vector of \( \text{RIT}_i \). Then, the function below is employed to classify a candidate segment \( cs \) using the acquired \( \alpha \):

\[
f(cs) = \text{sign} \left( \sum_{i=1}^{M} \alpha_i y_i \phi(\text{RIT}_i) \cdot \phi(\text{RIT}_i) + b \right), \tag{6}
\]

where \( \text{sign}() \) is the sign function which returns the sign of a number and \( b = y_1 - \left( \sum_{i=1}^{M} \alpha_i y_i \phi(\text{RIT}_i) \cdot \phi(\text{RIT}_i) \right) \) for any \( \text{RIT}_i \) such that \( \alpha_i = 0 \). As shown in Equations 5 and 6, the core of the SVM training and classification is the inner product of feature vectors, which measures the similarity of RITs under a high-dimensional vector space. A conventional vector representation of parse trees considers each possible tree fragment a vector dimension [10] such that the \( l \)-th component in \( \phi(\text{RIT}_i) \) is the occurrence frequency of the \( l \)-th tree fragment in \( \text{RIT}_i \). Based on this, the inner product determines the degree of overlap of two RITs. A problem with the vector representation is that the set of all possible tree fragments (i.e., the vector dimension) is huge, making the inner product computationally expensive [10]. Here, we leverage the following normalized convolution tree kernel (NCTK) [28] to compute the inner product of RITs in an efficient manner:

\[
K_{\text{NCTK}}(\text{RIT}_i, \text{RIT}_j) = \phi(\text{RIT}_i) \cdot \phi(\text{RIT}_j) = \frac{\sum_{n \in N} \sum_{n' \in N'} \Delta(n, n')}{\sqrt{\sum_{n \in N} \sum_{n' \in N'} \Delta(n, n') \cdot \sum_{n \in N} \sum_{n' \in N'} \Delta(n, n')}}, \tag{7}
\]

where \( N \) and \( N' \) are the sets of nodes in \( \text{RIT}_i \) and \( \text{RIT}_j \), respectively, and \( \Delta(n, n') \) is the number of the common subtrees rooted at \( n \) and \( n' \) that can be computed by the following recursive procedure:

1. If the production at \( n \) (i.e., \( n \) and all of its children) and the production at \( n' \) are different, then \( \Delta(n, n') = 0 \);
2. else if both \( n \) and \( n' \) are pre-terminals (i.e., POS tags), then \( \Delta(n, n') = 1 \);
3. else

\[
\Delta(n, n') = \prod_{i \in (\text{ch}(n, l), \text{ch}(n', l))} (1 + \Delta(\text{ch}(n, l), \text{ch}(n', l))), \tag{8}
\]

where \( |\text{ch}(n)| \) denotes the number of \( n \)'s children, and \( \text{ch}(n, l) \) is the \( l \)-th child of node \( n \). By using the normalized convolution tree kernel, the inner product of the RITs can be done in \( O(|N| \cdot |N'|) \).

It is noteworthy that SPIRIT extends our previous work CK [5] in a number of aspects. First, SPIRIT is a pure tree kernel method, whereas CK employs a composite kernel that integrates a tree kernel with a bigram kernel. Second, SPIRIT considers a clause as a basic segment unit, while the basic unit of CK is a sentence. Third, SPIRIT generalizes candidate segments into six structural types to facilitate the learning of representative person interactions. Fourth, SPIRIT enhances the pruning and ornamenting operators by using the proposed person dependency path.Clauses and the corresponding syntactical elements will be pruned if they are not associated with the person dependency path. Furthermore, the verbs in the path that convey interactive context will be identified to decorate our rich interactive tree with interactive semantics. Fifth, SPIRIT’s ornamenting operator examines the concepts of verbs and measures verb similarity by means of a concept taxonomy. By contrast, CK’s ornamenting simply employs a word-matching approach that will overlook many interactive segments. Through the above enhancements, SPIRIT not only prunes syntactical elements that are not associated with person interactions, but also skips irrelevant verbs that mislead the RIT ornamenting. Consequently, SPIRIT significantly outperforms CK.

### 4 EXPERIMENTATION

#### 4.1 Evaluation Dataset and Experiment Setting

To the best of our knowledge, there is no official corpus for person interaction detection. The entity relations defined in the Automatic Content Extraction (ACE) datasets, such as capital of, are static and irrelevant to person interactions. In [5], we compiled a small data corpus for a pilot study of person interaction detection. In this research, we extend the corpus to comprise 24 important topics from 2004 to 2014. Each topic consists of 100 Chinese news documents collected from Yahoo News. We employed the Stanford NER [29] to tag person names mentioned in the topic documents. The tagging produced 15,370 person names that represented 665 unique persons. We noticed that many of the person names rarely occurred in the topic documents and the rank-frequency distribution of the person names

---

followed Zipf’s law [21]. Names with low frequency usually refer to persons irrelevant to the topic (e.g., journalists). To reduce their influence on system performance, for each topic, we evaluated frequent person names whose frequency reached 70% of the total person name frequency in the topic documents. All of the evaluated person names thus represent important topic persons. The candidate segment generation algorithm extracted 9,632 candidate segments from the topic documents. Among them, 4,019 segments were labeled as interactive by two linguistic experts. The Kappa statistic of the labeling process is 0.845, which indicates that our corpus is reliable. As shown in Table 1, a great portion (i.e., 56.4%) of interactive segments are intra-clausal, and around half (49.8%) of the intra-clausal segments are non-interactive. In other words, persons occurring in the same clauses generally have no interaction, and the interactions of persons are usually narrated by a sequence of clauses. The distributions reveal that mining person interactions is not trivial. The corpus has been released to promote further research and is available at National Taiwan University Person Interactions Corpus (NTUPIC).

We evaluated the performance of interaction detection in terms of the precision rate, recall rate, and the F1 score [30]. In general, there is a trade-off between precision and recall. Since the two metrics evaluate system performance from different perspectives, a single metric that balances (averages) the trade-off is essential [31]. F1 is the harmonic mean of precision and recall. It is generally close to the minimum of the two values [21], and can thus be considered as an attempt to find the best possible compromise (balance) between precision and recall [31]. It is also deemed to be a conservative metric that prevents the possible overestimate of system performance because the harmonic mean is always less or equal to the arithmetic mean and geometric mean [21]. For this reason, F1 is extensively used to judge the superiority of information systems [31]. To derive credible evaluation results, we utilized the leave-one-topic-out or the so-called K-fold cross validation approach [32], in which K is 24 in this study. We evaluated the system performance over 24 runs, and in each run, all segments of a selected topic were used for testing, and the remaining 23 topics were used for the RIT classifier training. The testing results across all topics (i.e., 9,632 candidate segments) were then averaged to obtain the global system performance. Note that we applied LLR (i.e., Equation 1) to the training topics so as to construct a set of interactive verbs which were used for IV tagging. As each validation run excludes all segments of the testing topic from training, it successfully prevents LLR and the ornamenting process from seeing the testing information. This means that unbiased performance evaluations can be conducted. With respect to the convolution kernels of the RIT, Moschitti’s tree kernel toolkit [16] was adopted and implemented into the SVM Light package [11] for RIT classification.

### 4.2 System Component Evaluation

#### 4.2.1 Performance Evaluation of Interaction Detection

It is noteworthy that SPIRIT has a single parameter $\zeta$, which is the concept similarity threshold used by the RIT ornamenting to tag interactive semantics on candidate segments. We first examined its effect by setting the parameter between 0 and 1, and increasing it in increments of 0.1. Fig. 11 shows the interaction detection performance under different parameter settings. The precision and recall rates generally increase as $\zeta$ increases. This is because a large $\zeta$ makes the segments tagged with an IV tag be highly associated with person interactions. The setting thus improves the detection of interactive segments. By contrast, a small $\zeta$ would mislead the RIT ornamenting, thereby decreasing the interaction detection performance. However, overly constraining the similarity threshold (i.e., $\zeta > 0.7$) will decrease the recall, even though it increases the precision. This is because ornamenting upon a large threshold omits many synonyms of interactive verbs, thus many interactive segments were overlooked, resulting in a decreased recall rate. As observed in Fig. 11, setting $\zeta$ at 0.7 produces the best detection performance, and this setting was utilized in subsequent evaluations.

![PRF curve with different similarity thresholds.](image)  

Fig. 11. PRF curve with different similarity thresholds.
because the generalization process distinguishes candidate segments into six structure types that regularize text structures of person interactions. The generalization thereby facilitates the learning of representative interaction classification for each structural type, increasing the recall rate and the F score.

![Image](image_url)

**TABLE 2**

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F-score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIT without SSG</td>
<td>71.76</td>
<td>49.44</td>
<td>58.55</td>
</tr>
<tr>
<td>RIT with SSG</td>
<td>68.89</td>
<td>61.56</td>
<td>65.02</td>
</tr>
</tbody>
</table>

We then investigated the effect of the RIT that polishes the SPT by the operators of branching, pruning, and ornamenting. Here, the segment structure generalization is adopted due to its superior performance. Table 3 presents the performance of the SPT and the results of incrementally applying RIT branching, pruning, and ornamenting, denoted as +RITbranching, +RITpruning, and +RITornamenting, respectively. It is evident from the table that the RIT performs better than the SPT by simply applying the branching operator (i.e., +RITbranching) for intra-inner and inter-surrounding structural types with the error rate of SPIRIT being around 30%. We observed from the results that a great portion of false positives (i.e., non-interactive segments incorrectly recognized as interactive segments) were due to the irrealis mood that expresses plausible person interactions. For example, SPIRIT incorrectly classifies "歐巴馬想要延攬羅姆尼入閣(Obama wants to recruit Romney to join the cabinet)" as an interactive segment, even though the interaction mentioned in the segment did not happen yet.

For inter-clausal segments, a relatively high proportion of errors were false negatives. We noticed that the interactive verbs of inter-clausal segments were sometimes excluded from the generated person dependency paths. This may have been caused by the complex syntactic structures of the inter-clausal candidate segments, which mislead the dependency parsing process. It can be observed in the inter-clausal interactive segment "歐巴馬兌現承諾，在辦公室旁邊的總統專用餐廳和羅姆尼來場大和解午餐(Obama honors his promise to have a lunch reconciliation with Romney at the Presidential restaurant next to the President's office)", in which the interactive verb "和解(there is a reconciliation)" is absent from the generated person dependency path "歐巴馬...

![Image](image_url)

**TABLE 3**

Incremental Contribution of the RIT Branching, Pruning, and Ornamenting Operators with Six Structural Types

<table>
<thead>
<tr>
<th>RIT Structure</th>
<th>SPT</th>
<th>+RITbranching</th>
<th>+RITpruning</th>
<th>+RITornamenting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision, Recall, F1-scores (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intra-inner</td>
<td>74.53/97.00/84.29</td>
<td>74.53/97.00/84.29</td>
<td>74.85/98.65/85.12</td>
<td>77.06/91.65/83.72</td>
</tr>
<tr>
<td>Inter-inner</td>
<td>55.63/55.79/55.71</td>
<td>55.63/55.79/55.71</td>
<td>59.13/49.02/53.60</td>
<td>71.23/55.06/62.11</td>
</tr>
<tr>
<td>Intra-surrounding</td>
<td>68.76/85.17/76.09</td>
<td>66.45/90.97/76.80</td>
<td>66.98/92.36/77.65</td>
<td>72.94/88.35/79.91</td>
</tr>
<tr>
<td>Inter-surrounding</td>
<td>25.41/09.03/83.86</td>
<td>24.30/07.77/04.97</td>
<td>60.28/38.82/38.99</td>
<td>59.23/41.56/48.85</td>
</tr>
<tr>
<td>Intra-outer</td>
<td>43.12/25.62/32.15</td>
<td>49.08/31.84/38.62</td>
<td>60.79/31.09/41.14</td>
<td>61.98/47.24/53.62</td>
</tr>
<tr>
<td>Inter-outer</td>
<td>40.06/15.28/22.13</td>
<td>55.73/17.84/27.03</td>
<td>61.32/20.39/30.60</td>
<td>54.67/35.67/43.17</td>
</tr>
<tr>
<td>Micro-average</td>
<td>62.88/45.34/52.69</td>
<td>62.63/47.31/53.90</td>
<td>66.53/55.51/60.52</td>
<td>68.89/61.56/65.02</td>
</tr>
</tbody>
</table>

To summarize, the RIT branching operator enhances the SPT by extending the valuable segment context that is critical for person interaction detection. Next, the pruning operator removes indiscriminate and duplicate elements to make the RIT concise and clear. Finally, the RIT ornamenting highlights verbs closely associated with person interactions. As the operators polish the SPT from different perspectives, they do not conflict with each other. Consequently, applying the operators altogether achieves the best performance.

Based on our further analysis of the detection performance, Table 4 presents the detection errors over each structural type with the error rate of SPIRIT being around 30%. We observed from the results that a great portion of false positives (i.e., non-interactive segments incorrectly recognized as interactive segments) were due to the irrealis mood that expresses plausible person interactions. For example, SPIRIT incorrectly classifies "歐巴馬想要延攬羅姆尼入閣(Obama wants to recruit Romney to join the cabinet)" as an interactive segment, even though the interaction mentioned in the segment did not happen yet.
only occur after the last topic person. We noticed that our branching operator sometimes failed to extend a useful context because the last topic person name and the verbs following it could not form a verb phrase. Consequently, SPIRIT incorrectly recognized interactive segments as non-interactive segments. On the other hand, our method produces better performance on inner segments because their syntactic structures are relatively simple. Hence, since the generated person dependencies are concise, detection of person interactions is simplified.

Note that our segment structure generalization dropped 524 candidate segments (i.e., Others) as the segments having no verb occurring between or behind the topic persons. The linguistic experts, however, labeled some of them as interactive segments because the verbs that occurred before the topic persons conveyed person interactions. For instance, in the interactive segment "In 辯論會上批判歐巴馬的羅姆尼（Romney criticizes Obama in the debate）", the verb "批判（criticize）" associates Obama and Romney, and expresses their interaction. As our RIT is based on the shortest path connecting the topic persons in a segment’s parse tree, it cannot cover the verbs occurring before the topic persons, resulting in all of these segments being judged as non-interactive segments. Fortunately, this syntactic format is not common, which is why we only produced 76 false negatives.

<table>
<thead>
<tr>
<th>Structural Type</th>
<th>False Positive</th>
<th>False Negative</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra-inner (851)</td>
<td>177</td>
<td>55</td>
<td>27.26%</td>
</tr>
<tr>
<td>Inter-inner (1208)</td>
<td>126</td>
<td>255</td>
<td>31.54%</td>
</tr>
<tr>
<td>Intra-surrounding (1103)</td>
<td>242</td>
<td>88</td>
<td>29.92%</td>
</tr>
<tr>
<td>Inter-surrounding (3878)</td>
<td>355</td>
<td>764</td>
<td>28.86%</td>
</tr>
<tr>
<td>Intra-outter (1785)</td>
<td>198</td>
<td>369</td>
<td>31.76%</td>
</tr>
<tr>
<td>Inter-outter (283)</td>
<td>25</td>
<td>62</td>
<td>30.74%</td>
</tr>
<tr>
<td>Others (524)</td>
<td>0</td>
<td>76</td>
<td>14.50%</td>
</tr>
</tbody>
</table>

4.2.2 Comparison with Other Systems

Our research is closely related to RE and PPI. Here, we compare SPIRIT with 10 well-known RE and PPI methods: the tree kernel-based PPI method (TK-PPI) [33] in which the shortest dependency path between the topic persons in a segment’s parse tree is used for interaction detection, the shortest dependency path-directed constituent parse tree PPI method (SDP-CPT) [7] in which the tree representation generated from a constituent syntactic parser is refined by using the shortest dependency path between two topic persons derived from a dependency parser, the Chinese open relation extraction method (CORE) [34] which analyzes syntactic structures of text to extract entity relation, a composite kernel method that integrates the shortest path tree structure with an entity kernel for RE task (CKRE) [6], the SVM-based biomedical trigger extraction method (SVM-BTE) [35] which utilizes syntactic and context features to extract words regarding a certain biomedical event, the CRF-based relation descriptors extraction method (CRF-RDE) [36] which employs techniques of CRF to label the tokens in a text segment that describe a specific entity relation, and our previous work CK [5]. As mentioned in Section 3, CK integrates a tree kernel with a bigram kernel. SPIRIT, on the other hand, is a pure tree kernel method equipped with advanced rich interactive tree construction techniques (e.g., the new ornamenting and pruning operators, and the person dependency path). We also evaluate a variant of CK in which the bigram kernel is removed. We denoted CK without the bigram kernel as CKw. Comparing SPIRIT with CKw enables us to verify the contribution of the proposed RIT techniques. We further compare SPIRIT with the semantically smoothed tree kernel (SPTK) [37], [38] and its compositional variant CSPTK [39] using two dependency tree structures (denoted as SPTKw, SPTKw, CSPTKw, and CSPTKw). A merit of SPTK and CSPTK is that they measure the semantic similarity between nodes of dependency trees in terms of distributional analysis (e.g., Singular Value Decomposition). This enables structural similar trees to have a high similarity even if their nodes are associated with different but related labels. We adopted the KeLP package [40] to implement SPTK and CSPTK for comparison, and the Sinica Corpus was selected for their distributional analysis. In addition to these methods, two classification baselines including the Naive Bayes text classification method (NB) [21] and the support vector machine text classification method (SVM) [21] are evaluated to show the complexity of interaction detection. NB is a probabilistic classification based on Bayes’ theorem. The SVM text classification method represents a segment as a TF-IDF term vector [21] and trains a classifier by searching the hyperplane (i.e., decision boundary) that maximally separates the interactive and non-interactive training vectors.

In order to conduct a fair comparison, all 14 methods used for comparison are optimized by means of a unified parameter setting procedure against the evaluated dataset. Due to the space limitation, the procedure is presented at our data corpus website. Following are the optimized parameter settings. Except CORE, NB, and CRF-RDE, all of the methods compared are based on SVM classification. The SVM classification has a regularization parameter C which prevents overfitting of classifier training. In this comparison, C is set at 0.125, 0.25, 4, 2, 4, 2, 1, 0.5, 0.25, 2, 0.5, and 2 for SVM, SVM-BTE, TK-PPI, SDP-CPT, CKRE, CK, CKw, SPTKw, SPTKw, CSPTKw, and SPIRIT, respectively. The decay parameter $k$ of the convolution tree kernel-based methods (i.e., TK-PPI, SDP-CPT, CKw, and SPIRIT) is set at 0.4. In addition to C and $k$, CK and CKRE have two extra parameters $d$ and $\alpha$ for their composite kernel setting. The two parameters of CK and CKRE are set at $<2, 0.7>$ and $<2, 0.6>$, respectively. The decay parameters $\lambda$ and $\mu$ of SPTKw, SPTKw, CSPTKw, and CSPTKw are set at $<0.6, 0.4>$, $<0.8, 0.4>$, $<0.3, 0.5>$, and $<0.6, 0.4>$. For CRF-RDE, the CRF model parameters $c$ and $f$ are set as 9 and 20, respectively. Note that the training and testing of the Na-
iveBayes text classification method are based on term frequencies only and thus has no parameter tuning. Similarly, the syntactic rule-based method CORE adopts a set of heuristic rules to extract person interactions from syntactic structure and also does not require parameter tuning.

TABLE 5

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>42.90% (**)</td>
<td>40.29% (**)</td>
<td>41.55% (**)</td>
</tr>
<tr>
<td>SVM</td>
<td>46.21% (**)</td>
<td>48.32% (**)</td>
<td>47.24% (**)</td>
</tr>
<tr>
<td>TK-PPI</td>
<td>53.88% (**)</td>
<td>41.19% (**)</td>
<td>46.69% (**)</td>
</tr>
<tr>
<td>SDP-CPT</td>
<td>50.91% (**)</td>
<td>52.30% (**)</td>
<td>51.60% (**)</td>
</tr>
<tr>
<td>CORE</td>
<td>46.53% (**)</td>
<td>46.75% (**)</td>
<td>46.64% (**)</td>
</tr>
<tr>
<td>CKRE</td>
<td>53.55% (**)</td>
<td>49.86% (**)</td>
<td>51.64% (**)</td>
</tr>
<tr>
<td>SVM-BTE</td>
<td>43.70% (**)</td>
<td>78.12% (-)</td>
<td>63.05% (**)</td>
</tr>
<tr>
<td>CRF-RDE</td>
<td>63.54% (**)</td>
<td>43.59% (***)</td>
<td>51.71% (**)</td>
</tr>
<tr>
<td>CK</td>
<td>63.56% (**)</td>
<td>51.60% (**)</td>
<td>56.96% (**)</td>
</tr>
<tr>
<td>CK</td>
<td>71.33% (-)</td>
<td>54.23% (**)</td>
<td>61.62% (**)</td>
</tr>
<tr>
<td>SPTK</td>
<td>60.57% (**)</td>
<td>55.55% (**)</td>
<td>57.95% (**)</td>
</tr>
<tr>
<td>SPTK</td>
<td>61.36% (**)</td>
<td>55.07% (**)</td>
<td>58.04% (**)</td>
</tr>
<tr>
<td>CSPTK</td>
<td>60.18% (**)</td>
<td>56.47% (**)</td>
<td>58.27% (**)</td>
</tr>
<tr>
<td>CSPTK</td>
<td>62.92% (**)</td>
<td>55.89% (**)</td>
<td>59.20% (**)</td>
</tr>
<tr>
<td>SPIRIT</td>
<td>68.89%</td>
<td>61.56%</td>
<td>65.02%</td>
</tr>
</tbody>
</table>

A one-tail paired z-test [41] is applied to determine whether SPIRIT significantly improve the performance of the comparisons, where *, **, and *** represent z-tests with α = 0.1, 0.05, and 0.01, respectively.

Table 5 displays the comparison results. The F1 scores of NB and SVM are generally worse than those of the compared methods. This is because the two methods are simply based on tokens (words) and frequencies, neglecting the syntactic and semantic information of the text segments that other methods draw upon. The results also indicate that interaction detection is more difficult than the traditional text classification task. While TK-PPI examines interactive verbs to detect interactive segments, it is based on the complete parse tree which is generally long and comprised of redundant elements. By contrast, CKRE utilizes the SPT to refine the syntactic structure of a candidate segment. Hence, it surpasses TK-PPI by about 5% in terms of the F1 score. SDP-CPT further employs a syntactic dependency parser to polish the parse tree of a candidate segment, especially those of inter-clausal segments, which are superior to those of baseline and TK-PPI. SPIRIT significantly outperforms CKRE due to the rich interactive tree constructed by SPIRIT being much more informative than that by CK. Our superior precision indicates that the proposed person dependency path is effective in pruning indiscriminative clauses. The performance results closely correspond with the improved precision from our pruning operator shown in Table 2. SPIRIT’s superior recall validates the effectiveness of our ornamenting operator in that considering the concepts of verbs enables the detection of diverse person interactions. CK achieves a high precision when the bigram kernel is employed. The bigram kernel examines the content similarity of the candidate segments and the interactive segments in the training dataset. As the detected segments are similar to the interactive segments, their detections are correct. However, given that the bigram kernel overlooks the concept of words, the recall of CK is inferior to ours.

The precision rate of CORE was inferior, and the reason for this is that the method regards the verbal phrase to exist in the parse tree of a segment as a cue of person interactions. In the experiment dataset, a considerable number of non-interactive segments contained verbal phrases, causing the method to generate many false positives that decreased its precision rate. On the other hand, the recall of SVM-BTE was high due to the ability of the method to explore the bigrams and the trigrams of interactive verbs to detect interactive segments. Since Chinese keywords are generally compounds [42], many of the bigrams and the trigrams also convey interaction semantics. As a consequence, the method detects many interactive segments to increase the recall rate. Unfortunately, SVM-BTE overlooks the person dependency path, so many of the bigrams and the trigrams that occurred in the detected segments were unrelated to the topic persons. As a result, the method generated a lot of false positives that impaired its detection precision. It is worth noting that the precision of CRF-RDE was high. The method employs stringent feature functions which examine the context of entities in a given text, and the detected interactive segments are reliable and correct. For instance, in addition to locating an interactive verb in a candidate segment, many contextual feature functions further check the surrounding words and the corresponding POS tags to activate the feature functions. The feature functions, however, are too rigid, and this led to a low recall rate and poor system performance.

As shown in Table 5, our method outperforms SPTK and CSPTK significantly. Based on further analysis of the detection performance, we observed that the SPTK and CSPTK methods did not excel on inter-clause candidate segments that normally contain complex syntactic structures. For instance, the precision, recall, and F1 scores of SPTK_LCT for the inter-clause candidate segments are 52.31, 48.20, and 50.17, respectively, which are relatively lower than their overall performance. The inferior performance is due to the dependency tree structures (e.g., LCT and GRCT) used in the SPTK and CSPTK methods, which contain a great amount of redundant elements irrelevant to person interactions. By contrast, our pruning operator is able to eliminate indiscriminative and redundant syntactic elements to help our method learn representative RIT patterns of person interactions. We also noticed that the document corpus used for distributional analysis (i.e. the Sinica corpus) is relatively smaller than that used in [37], [38], [39],
even though it is one of the biggest Chinese corpora. As reported in [43], the performance of the semantically smoothened tree kernel depends on the quality of the input document corpus. The lack of comprehensive Chinese document corpora may also contribute to the inferior performances of SPTK and CSPTK.

Finally, we evaluate the performances of the compared methods using the 11-point precision recall curves [31]. To plot the curves, the evaluated candidate segments were sorted according to their classification scores. Note that the classifications of CRF-RDE and SVM-BTE are based on a binary criterion, so the classification results cannot be graphed. Fig. 12 shows that the precisions of our method at the 11 recall levels are superior to those of the compared methods. In other words, our method is able to detect interactive segments accurately.

To summarize, SPIRIT achieves a remarkable interaction detection performance by means of a pure and concise tree kernel. The proposed rich interactive tree structure successfully integrates the syntactic and semantic information of a text, and further prunes indiscriminative syntactic elements and extends the informative segment context. Consequently, our method significantly outperforms the compared methods.

5 CONCLUSION REMARKS

A topic is associated with specific times, places, and persons. Knowing the interactions between the topic persons can help readers construct the background of the topic and facilitate document comprehension. In this paper, we have proposed a topic person interaction detection method called SPIRIT, which identifies text segments mentioning person interactions in a set of topic documents. We designed a rich interactive tree structure which employs three RIT operators to explore the syntactic and semantic information in a text. The information is then applied to the convolution tree kernel to classify the interactive segments. The experiment results based on real-world datasets demonstrate that the proposed RIT operators can successfully exploit the syntactic structures, interaction semantics, and segment context relevant to person interactions. Consequently, our method outperforms many well-known relation extraction and tree kernel-based PPI methods.

Current results have paved the way for other potential research topics. For instance, we observed that person interactions generally involve sentiments. The sentiment information of a text can be investigated to enhance our rich interactive tree structure and to improve the interaction detection results. Furthermore, developing effective interaction keyword extraction methods would be worthwhile in order to better represent person interactions. In our performance evaluation, CK’s precision shows the potential of content similarity for interaction detection. Instead of compositing another content kernel (e.g., the bigram kernel), we are designing a new RIT ornamenting operator in which tokens in the person dependency path will be examined to measure the content similarity between RITs. Moreover, concepts of the tokens will be incorporated to increase both detection precision and recall. We will also employ techniques of word sense disambiguation to improve the verb similarity calculation of RIT ornamenting. Finally, SPIRIT is adaptable to different languages. Taking English as an example, we can detect English person interactions by adopting the English version of Stanford parser which labels dependencies between English tokens, and replacing E-HowNet with an English lexical database (e.g., WordNet) to calculate verb similarity. In the future, we will also deploy SPIRIT for English texts.

ACKNOWLEDGMENT

This research was supported by the Ministry of Science and Technology of Taiwan under grant MOST 103-2221-E-002-106-MY2 and MOST 103-3111-Y-001-027.

REFERENCES


Yung-Chun Chang received an M.S. degree in information management from Chiang Jung Christian University, Taiwan in 2007. He is currently working toward a Ph.D. degree in information management at the National Taiwan University, Taiwan. His research interests include information retrieval and text mining.

Chien Chin Chen received a Ph.D. degree in electrical engineering from National Taiwan University, Taiwan, in 2007. He is currently an assistant professor in the Department of Information Management at National Taiwan University. His papers have appeared in IEEE Transactions on Knowledge and Data Engineering (TKDE), ACM Transactions on Information Systems (TOIS), SIGIR, SIGKDD, COLING, etc. His current research interests include text mining, information retrieval, and knowledge discovery.

Wen-Lian Hsu received a Ph.D. degree in operations research from Cornell University in 1980. He is currently the Director and a Distinguished Research Fellow of the Institute of Information Science, Academia Sinica, Taiwan. His earlier contribution was on graph algorithms and he has applied similar techniques to tackle computational problems in biology and natural language. In 1993, he developed GOING, which has since revolutionized Chinese input on computer. Dr. Hsu received the Outstanding Research Award from the NSC in 1991, 1994, 1996, the first K. T. Li Research Breakthrough Award in 1999, the IEEE Fellow in 2006, and the Teco Award in 2008.