A Composite Kernel Approach for Detecting Interactive Segments in Chinese Topic Documents

Yung-Chun Chang[1,2], Chien Chin Chen[1], and Wen-Lian Hsu[2]

[1] Department of Information Management, National Taiwan University
No. 1, Sec. 4, Roosevelt Rd., Taipei City 10617, Taiwan (R.O.C)
[2] Institute of Information Science, Academia Simica
No. 128, Sec. 2, Academia Rd., Taipei City 11529, Taiwan (R.O.C)
{changyc,hsu}@iis.sinica.edu.tw, patonchen@ntu.edu.tw

Abstract. Discovering the interactions between persons mentioned in a set of topic documents can help readers construct the background of a topic and facilitate comprehension. In this paper, we propose a rich interactive tree structure to represent syntactic, content, and semantic information in text. We also present a composite kernel classification method that integrates the tree structure with a bigram kernel to identify text segments that mention person interactions in topic documents. Empirical evaluations demonstrate that the proposed tree structure and bigram kernel are effective and the composite kernel approach outperforms well-known relation extraction and PPI methods.

Keywords: Topic Mining, Interaction Detection, Rich Interactive Tree, Composite Kernel.

1 Introduction

The web has become a powerful medium for disseminating information about diverse topics, such as political issues and sports tournaments. While people can easily find documents that cover various perspectives of a topic, they often have difficulty assimilating the information in large documents. The problem has motivated the development of several topic mining methods to help readers digest enormous amounts of topic information. For instance, Nallapati et al. [13] and Feng and Allan [5] grouped topic documents into clusters, each of which represents a theme in a topic. The clusters are then connected chronologically to form a timeline of the topic. Chen and Chen [1] developed a method that summarizes the incidents of a topic's timeline to help readers quickly understand the whole topic. The extracted themes and summaries distill the topic contents clearly; however, readers still need to expend a great deal of time to comprehend the extracted information about unfamiliar topics.

Basically, a topic is associated with specific times, places, and persons [13]. Thus, discovering the interactions between persons mentioned in topic document can help readers construct the background of the topic and facilitate comprehension. For instance, if readers know the interactions of the key persons in a presidential campaign, they can understand documents about the campaign more easily. Interaction discovery is an active

R.E. Banchs et al. (Eds.): AIRS 2013, LNCS 8281, pp. 215–226, 2013.

research area in the bioinformatics field. A number of studies [e.g., 15, 18] have investigated the problem of protein-protein interaction (PPI) which focuses on discovering the interactions between proteins mentioned in biomedical literature. Specifically, discovering PPIs involves two major tasks: *interaction detection* and *interaction extraction* [10]. The first task decomposes medical documents into text segments and identifies the segments that convey interactions between proteins. Then, the second task applies an information extraction algorithm to extract interaction tuples from the identified segments. In this paper, we focus on interaction detection in Chinese topic documents and identify text segments (called *interactive segments* hereafter) that convey interactions between persons. According to [17], such interactions exemplify types of human behavior that make people consider each other or influence each other. Examples of person interactions include compliments, criticism, collaboration, and competition. The detection of interactions between topic persons is more difficult than the task in PPI because the latter tries to discover permanent interactions between proteins, such as binding. By contrast, the interactions between persons are dynamic and topic-dependent. For instance, during the 2012 U.S. presidential election, the Democratic candidate, incumbent President Barack Obama often criticized Mitt Romney (the Republican candidate) for his political views. However, in the topic about Obama forming a new cabinet, President Obama broke bread with Mitt Romney at the White House, and even considered offering him a position in the new cabinet.

To detect interactive segments from topic documents effectively, we model interaction detection as a classification problem. We develop a rich interactive tree structure to represent syntactic, content, and semantic information in text segments. Furthermore, to identify interactive segments in topic documents, we develop a composite kernel classification method that integrates the tree structure with a bigram kernel to support vector machines (SVM) [7]. The results of experiments demonstrate that the composite kernel classification method is effective in detecting interactive segments. In addition, the proposed rich interactive tree structure and bigram kernel successfully exploits the syntactic structures, interaction semantics, and content of text segments. Consequently, the method outperforms the tree kernel-based PPI method [15]; the feature-based interaction detection method [2]; and the shortest path-enclosed tree (SPT) detection method [21], which is widely used to identify relations between named entities.

2 Related Work

Our research is closely related to relation extraction (RE), which was introduced as a part of the template element task in the sixth Message Understanding Conference (MUC-6). The goal of RE is to discover the semantic relations between the following five types of entities in text: persons, organizations, locations, facilities, and geo-political entities. Many RE methods [e.g., 4, 6, 8, 19] treat relation extraction as a supervised classification problem. Given a training corpus containing a set of manually-tagged examples of predefined relations, a supervised classification algorithm is employed to train an RE classifier to assign (i.e., classify) a relation type to a new text segment (e.g., a sentence). Feature-based approaches [6, 8] and kernel-based approaches [4, 19] are frequently used for RE. Feature-based methods exploit

instances of positive and negative relations in a training corpus to identify effective text features for relation extraction. For instance, Hong [6] applied a set of features that included lexical tokens, syntactic structures, and semantic entity types, to SVM for relation extraction. In addition, Kambhatla [8] integrated lexical, syntactic, and semantic features of text into a maximum entropy model to extract relations between entities in the Automatic Context Extraction (ACE) datasets[1]. Feature-based methods often have difficulty finding effective features to extract entity relations. To resolve the problem, Collins and Duffy [3] developed a convolution tree kernel (CTK) that computes the similarity between two text segments in terms of the degree of overlap between their constituent parsing trees. A relation type is assigned to a text segment if the segment is similar to instances of the relation type in the training corpus. Moschitti [12] also utilized a CTK in the predicate argument classification task, which is a special case of relation extraction. Zhang et al. [21] further refined the convolution tree kernel by using the shortest path-enclosed tree (SPT) structure, which is the sub-tree enclosed by the shortest path linking two entities in a parsing tree. Their experiment results showed that the SPT successfully represents syntactic information in text and therefore achieves a superior relation extraction performance on the ACE dataset. In recent years, a technique that combines CTK with SPT has been applied by many RE methods [20].

Our research is also related to the protein-protein interaction (PPI) detection [14], which focuses on discovering protein interactions mentioned in biomedical literature. In medical research, determining protein interaction pairs is crucial to understanding both the functional role of individual proteins and the organization of the entire biological process. Originally, methods on PPI are feature-based. The methods extract text features from sentences to construct learning models, which are then used to detect sentences that mention protein interactions. For instance, Ono et al. [14] manually defined a set of syntactic rule-based features covering word and part-of-speech patterns to represent complex sentences. Xiao et al. [18] exploited maximum entropy models to combine diverse lexical, syntactic, and semantic features for PPI extraction. However, the above features hardly represent structured and dependency-based syntactic information in a constituent, which is essential for detecting interactions between proteins. To address the issue, several tree-based kernel approaches have been developed. For example, Qian et al. [16] defined a set of hand-crafted heuristics to identify the informative parts of a constituent parsing tree. The identified sub-trees are then examined by a classification model, which assigns a relation type to the proteins mentioned in a text segment. Miyao et al. [11] combined constituent parsing trees with a bag-of-words kernel to improve PPI performance. Recently, Qian and Zhou [15] developed a novel tree-based kernel. In their approach, the parsing tree of text generated by a constituent syntactic parser is revised by the shortest dependency path between two proteins derived from a dependency parser. Their experiment results show that the tree-based kernel is efficient in PPI detection. Our research differs from RE and PPI, which detect static and permanent relations. In contrast, our research detects interactions between persons, which are dynamic and topic-dependent.

[1] http://www.itl.nist.gov/iad/mig/tests/ace/

3 The Composite Kernel Approach for Interaction Detection

Figure 1 shows the proposed interaction detection method, which is comprised of three key components: *candidate segment generation*, *rich interactive tree construction*, and *composite kernel classification*. We regard interaction detection as a classification problem. The candidate segment generation component processes a set of Chinese topic documents to extract text segments (called *candidate segments* hereafter) that may mention interactions between topic persons. Then, each candidate segment is represented by a rich interactive tree that integrates the syntactic, content, and semantic information extracted from the segment. Finally, the composite kernel classification component combines the rich interactive tree with a bigram kernel for SVM to classify interactive segments. We discuss each component in detail in the following sub-sections.

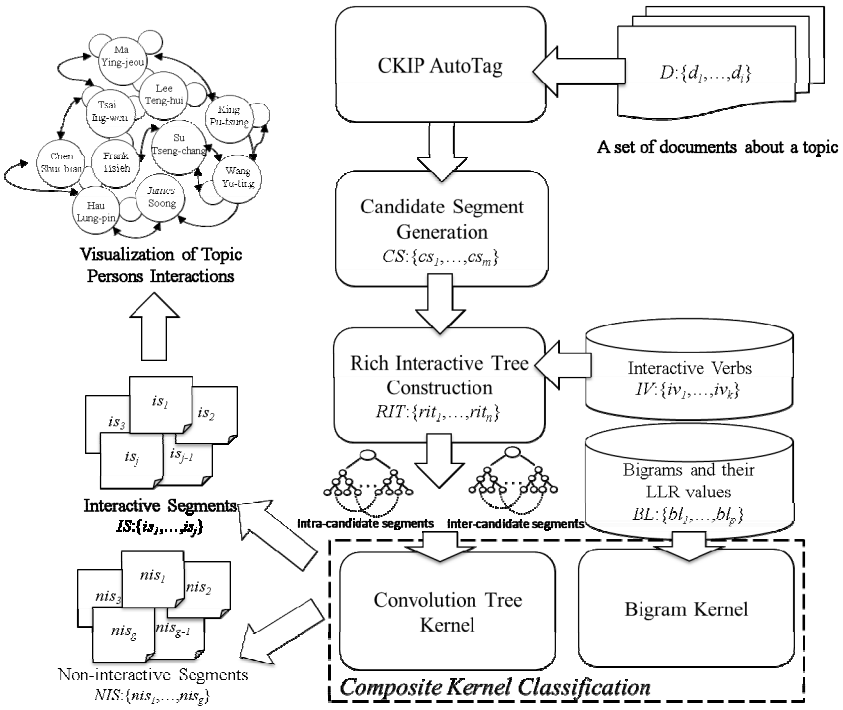

Fig. 1. The interaction detection method

3.1 Candidate Segment Generation Component

For a Chinese topic document *d*, we first apply the Chinese word segmentation system CKIP AutoTag[2] to decompose the document into a sequence of sentences $S = \{s_1,\ldots,s_k\}$.

[2] http://ckipsvr.iis.sinica.edu.tw/

CKIP also labels the tokens in the sentences that represent a person's name. We observed that the rank-frequency distribution of the labeled person names followed Zipf's law [9], which means that many of them rarely occurred in the topic documents. Low frequency names usually refer to persons that are irrelevant to the topic (e.g., journalists), so they are excluded from the interaction detection process. Let $P = \{p_1,\ldots,p_e\}$ denote the set of important topic person names. For any topic person name pair (p_i, p_j) in P, the candidate segment generation component extracts text segments that are likely to mention the pair's interactions from the document. The component processes a set of document sentences S one by one and considers a sentence as the initial sentence of a candidate segment if it contains person name p_i (p_j). Because the interaction between p_i and p_j may be narrated by a sequence of sentences, we consider two types of candidate segments, namely, *intra-candidate segments* and *inter-candidate segments*. The component then examines the initial sentence and subsequent sentences until it reaches an end sentence that contains the person name p_j (p_i). If the initial sentence is identical to the end sentence, the process generates an intra-candidate segment; otherwise, it generates an inter-candidate segment. However, if there is a period in the inter-candidate segment, we drop the segment because, in Chinese, a period indicates the end of a discourse. In addition, if p_i (p_j) appears more than once in an inter-candidate segment, we truncate all the sentences before the last p_i (p_j) to make the candidate segment concise. By running all person name pairs of P over the topic documents, we obtain a candidate segment set $CS = \{cs_1,\ldots, cs_m\}$.

3.2 Rich Interactive Tree Construction Component

A candidate segment is represented by the rich interactive tree (RIT) structure, which is the shortest path-enclosed tree (SPT) of the segment enhanced by three operators: *branching*, *pruning*, and *ornamenting*. To facilitate comprehension of the operators, the inter-candidate segment shown in Fig. 2(a), which mentions the interaction between "歐巴馬(Barack Obama)" and "羅姆尼(Mitt Romney)", serves as an example.

(1) RIT branching

In [21], the authors show that the SPT is effective in identifying the relation between two entities mentioned in a segment of text. Given a candidate segment, the SPT is the smallest sub-tree of the segment's syntactic parsing tree that links person names p_i and p_j, but the information in the SPT is often insufficient for interaction detection. For instance, in Fig. 2(a), "延攬(recruit)" and the corresponding syntactic constituent are critical for recognizing the interaction between Obama and Romney. However, they are excluded from the SPT, as shown in Fig. 2(b). To include useful segment context information, the branching operator extends the SPT by examining the syntactic structure of the candidate segments. By default, we utilize the SPT as our RIT sapling. However, if the last person name and the verb following it form a verb phrase in the syntactic parsing tree, we treat the verb as a modifier of the last person name and extend the RIT to the end of the verb phrase. As shown in Fig. 2(c), the extended RIT includes richer context information than the SPT.

(2) RIT pruning

To make the RIT concise and clear, we prune redundant elements via the following procedures.

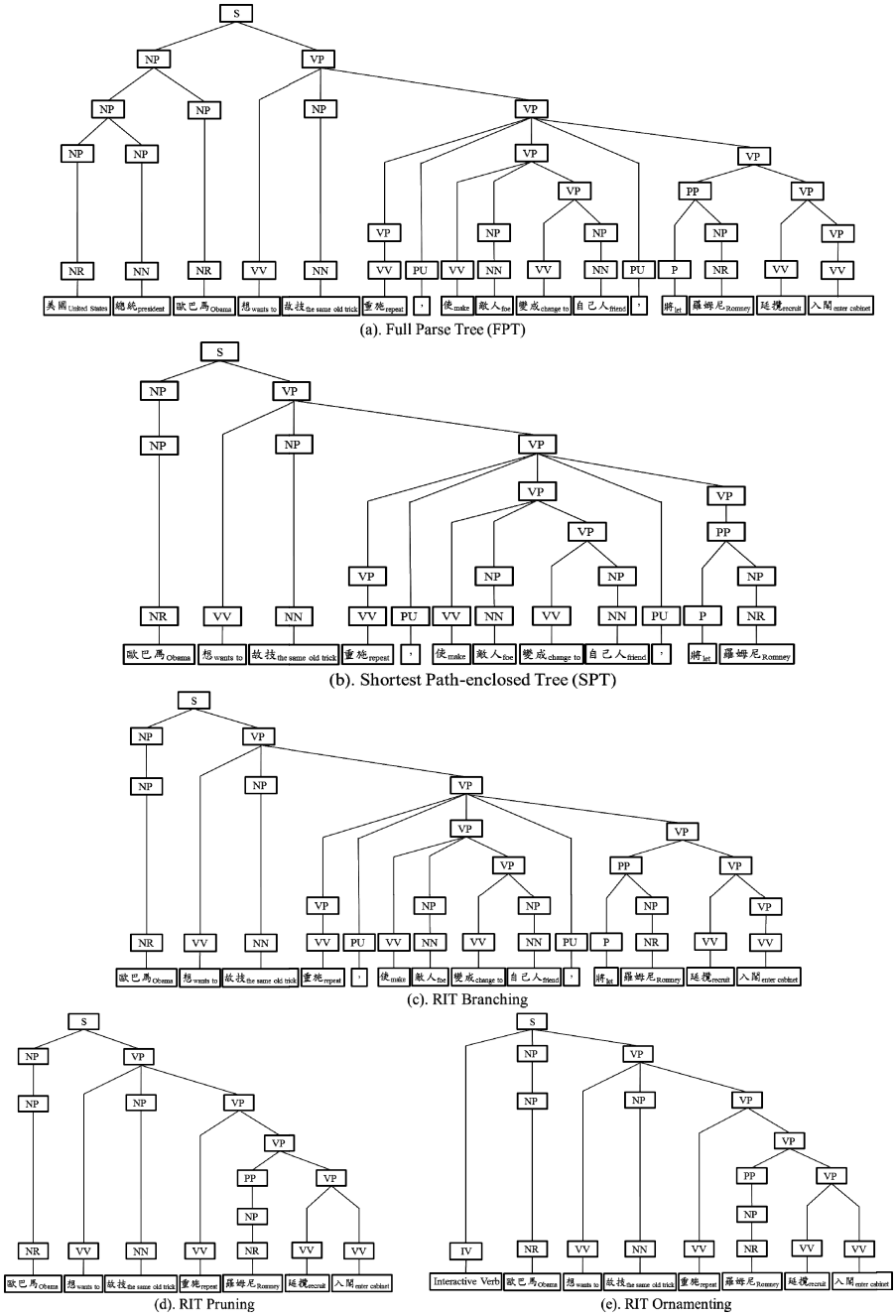

Fig. 2. The rich interactive tree construction procedure for an inter-candidate segment "美國總統歐巴馬想故技重施，使敵人變成自己人，將羅姆尼延攬入閣(United States President Obama wants to repeat the same old trick by changing from foe to friend and recruit Romney to join the cabinet)"

- Truncating inter-candidate segments: We observe that the middle sentences of inter-candidate segments do not normally contain information that can be used to detect interactions between persons. For instance, in Fig. 2(c), the middle sentence "使敵人變成自己人(changing from foe to friend)" is not useful for recognizing the interaction "延攬(recruit)" between Obama and Romney. In each inter-candidate segment, we remove all the middle sentences if the segment is composed of more than two sentences. The corresponding elements in the RIT and the punctuation are also deleted to concatenate the initial and end clauses.
- Removing indiscriminative RIT elements: Frequent words are not useful for expressing interactions between topic persons. For instance, the word "將(let)" in Fig. 2(c) is a common Chinese word and cannot discriminate interactive segments. To remove stop words and the corresponding syntactic elements from the RIT, we sort Chinese words according to their frequency in the text corpus. Then, the most frequent words are used to compile a stop word list. Moreover, to refine the list, person names and verbs are excluded from it because they are key constructs of person interactions.
- Merging duplicate RIT elements: We observe that nodes in an RIT are sometimes identical to their parents. For instance, the branch "重施$_{\text{repeat}}$→VV→VP→VP" in Fig. 2(c) contains two successive VP's. The tree-based kernel we use to classify a candidate segment computes the overlap between the RIT structure of the segment and that of the training segments. Because complex RIT structures degrade the computation of the overlap, we merge all duplicate elements to make the RIT concise.

(3) RIT ornamenting

Verbs are often good indicators of interactive segments, but not all verbs express person interactions. Highlighting verbs (called *interactive verbs* hereafter) closely associated with person interactions in an RIT would improve the interaction detection performance. We used the log likelihood ratio (LLR) [9], which is an effective feature selection method, to compile a list of interactive verbs. Given a training dataset comprised of interactive and non-interactive segments, the LLR calculates the likelihood that the occurrence of a verb in the interactive segments is not random. 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 the interactive verb list. For each RIT that contains an interactive verb, we add an IV tag as a child of the tree root to incorporate the interactive semantics into the RIT structure (as shown in Fig. 2(e)).

3.3 Composite Kernel

Kernel approaches are frequently used in SVM to compute a dot product (i.e., similarity) between instances modeled in a complex feature space. In this study, we employ a composite kernel approach that integrates the convolution tree kernel (CTK) [12] with a bigram kernel to determine the similarity between segments.

(1) Convolution Tree Kernel

We leverage the convolution tree kernel to capture the syntactic similarity between rich interactive trees. Specifically, the convolution tree kernel K_{CTK} counts the number of common sub-trees as the syntactic similarity between two rich interactive trees RIT_1 and RIT_2 as follows:

$$K_{CTK}(RIT_1, RIT_2) = \sum_{n_1 \in N_1, n_2 \in N_2} \Delta(n_1, n_2) \quad (1)$$

where $N1$ and $N2$ are the sets of nodes in $RIT1$ and $RIT2$ respectively. In addition $\Delta(n_1, n_2)$ evaluates the common sub-trees rooted at n_1 and n_2 and is computed recursively as follows:

(1) if the productions (i.e. the nodes with their direct children) at n_1 and n_2 are different, $\Delta(n_1, n_2) = 0$;

(2) else if both *n1* and *n2* are pre-terminals (POS tags), $\Delta(n_1, n_2) = 1 \times \lambda$;

(3) else calculate $\Delta(n_1, n_2)$ recursively as:

$$\Delta(n_1, n_2) = \lambda \prod_{k=1}^{\#ch(n_1)} (1 + \Delta(ch(n_1, k), ch(n_2, k))), \quad (2)$$

where $\#ch(n_1)$ is the number of children of node n_1; $ch(n, k)$ is the k^{th} child of node n; and $\lambda (0<\lambda<1)$ is the decay factor used to make the kernel value less variable with respect to different sized sub-trees.

(2) Bigram Kernel

In addition to the syntactic similarity, we consider the content similarity. Since most Chinese keywords are comprised of two characters, we design the following bigram kernel $K_{BK}(.)$, which examines the bigrams in a candidate segment *cs* and a training segment *ts* to measure their content similarity as follows:

$$K_{BK}(cs, ts) = \sum_i \sum_j C(cs \cdot b_i, ts \cdot b_j), \quad (3)$$

where b_i and b_j represent the i^{th} bigram of *cs* and j^{th} bigram of *ts* respectively. The function $C(.)$ returns the LLR value of $cs.b_i$ if $cs.b_i$ and $ts.b_j$ are identical; otherwise, it returns 0. As the LLR value of an interactive verb, a bigram's LLR value indicates the weight of bigram associated with interactive or non-interactive segments. Consequently, the value of K_{BK} will be high if the bigram overlap between *cs* and *ts* is large, and the overlapping bigrams are discriminative.

Finally, a composite kernel approach is used to interpolate the convolution tree kernel and the bigram kernel. We exploit polynomial interpolation [21], which integrates the two kernels as follows:

$$K_{COM}(cs, ts) = \alpha \cdot K_{BK}^{P}(cs, ts) + (1 - \alpha) \cdot K_{CTK}(RIT_{cs}, RIT_{ts}), \quad (4)$$

where *cs* denotes a candidate segment, *ts* is a training segment in the training corpus, and RIT_{cs} and RIT_{ts} are the corresponding rich interactive trees. $K^P(\bullet,\bullet) = (K(\bullet,\bullet) + 1)^d$ and it is the polynomial expansion of kernel $K(\bullet,\bullet)$. The parameters d and α are the polynomial degree and weight coefficient respectively.

4 Performance Evaluation

4.1 Experimental Setting

To the best of our knowledge, there is no official corpus for person interaction detection. The relations defined in the Automatic Context Extraction (ACE) datasets,

such as *capital of,* are static and irrelevant to person interactions. Therefore, we compiled a data corpus for the performance evaluations, as shown in Table 1. It contains 10 topics related to political events from 2004 to 2012; and each topic consists of 50 Chinese news documents (all longer than 250 words) collected from Yahoo News. As mentioned in Section 3, many of the person names labeled by CKIP rarely occur in topic documents, and low frequency names usually refer to persons that are irrelevant to the evaluated topics. Hence, for each topic, we evaluated the person names whose frequency reached 70% of the total person name frequency in the topic documents. All the evaluated names represent important topic persons. We used the candidate segment generation algorithm to extract 1754 candidate segments from the topic documents, and two experts labeled 651 of the segments as interactive. The Kappa statistic of the labeling process is 0.834, which means that our data corpus is reliable.

Table 1. The statistics of data corpus

# of topics	10
# of topic documents	500
# of tagged person names	332
# of evaluated person names	67
# of person name pairs	276
# of interactive segments (intra)	338
# of interactive segments (inter)	313
# of non-interactive segments (intra)	380
# of non-interactive segments (inter)	723

We use the SVMLight package [7] to implement our composite kernel classification component; and set the polynomial kernel parameters d and α at 2 and 0.23 respectively, as suggested in [20, 21]. In addition, we use Moschitti's tree kernel toolkit [12] to develop the convolution kernel of an RIT. To derive credible evaluation results, we utilize the leave-one-out cross validation method [9]. The evaluation metrics are the precision rate, recall rate, and F1-score [9]. The F1 value is used to determine relative effectiveness of the compared methods.

4.2 Results and Discussion

The proposed RIT structure uses three operators, *branching*, *pruning*, and *ornamenting*, to enhance the SPT. In the following, we evaluate the performance of the operators to demonstrate the effectiveness of RIT. Table 2 shows the marginal performances of applying RIT branching, pruning, and ornamenting, denoted as $+RIT_{branching}$, $+RIT_{pruning}$, and $+RIT_{ornamenting}$ respectively. In addition, to demonstrate the efficacy of the proposed method, we detail the results of applying our composite kernel classification component (denoted as RIT+BK), which integrates the RIT with the bigram kernel. As shown in the table, only RIT branching (i.e., $+RIT_{branching}$) outperforms the SPT. This is because the branching operator correctly extends useful

context information to remedy the context-limited problem of the SPT (see Sec. 3.2). The pruning operator further improves the system performance because it successfully eliminates indiscriminative and redundant RIT elements and thereby helps SVM learn representative syntactic structures of person interactions. The RIT ornamenting operator improves the F1 performance significantly. Moreover, the compiled interactive verbs are highly correlated with person interactions, so tagging them in the RIT structure helps our method discriminate interactive segments. Notably, our composite kernel classification component achieves the best performance. As the bigram kernel examines the content of segments to identify interactive segments, it does not conflict with the RIT, which analyzes syntactic and semantic information in the segments. Consequently, applying them together improves the system performance.

Table 2. Incremental contribution of the RIT branching, pruning, and ornamenting operators

RIT Structure	*Intra-segment*	*Inter-segment*	*Micro-average*	*Macro-average*
	Precision, Recall, F1-score (%)			
SPT	69.42 / 57.10 / 62.66	45.74 / 13.74 / 21.13	63.44 / 36.25 / 46.14	57.58 / 35.42 / 41.90
$+RIT_{branching}$	69.15 / 60.36 / 64.45	49.56 / 17.89 / 26.29	63.73 / 39.94 / 49.10	59.36 / 39.12 / 45.37
$+RIT_{pruning}$	76.76 / 64.50 / 70.10	43.66 / 18.81 / 27.25	62.70 / 43.59 / 51.43	60.21 / 42.16 / 48.68
$+RIT_{ornamenting}$	83.56 / 72.19 / 77.46	73.55 / 56.87 / 64.14	79.03 / 64.82 / 71.22	78.56 / 64.53 / 70.80
RIT + BK	**85.16 / 78.11 / 81.48**	**77.27 / 59.74 / 67.36**	**81.70 / 69.28 / 74.98**	**81.22 / 68.93 / 74.44**

In addition to the SPT and the proposed method, we evaluate FISER [2] and SDP-CPT [15]. To ensure the fairness of our evaluation, systems used for comparison are also developed using the SVMLight package [7] and Moschitti's tree kernel toolkit [12]. It has been shown that SPT is an effective relation extraction method [21]. FISER exploits nineteen features that cover parts-of-speech, context and semantic information in text to detect interactive segments in topic documents. SDP-CPT is an effective tree kernel-based PPI method that analyzes the syntactic dependency tree of a piece of text to identify protein interactions. In this paper, we use it to identify person interactions. As shown in Table 3, the proposed method significantly outperforms SPT and SDP-CPT. This is because SPT and SDP-CPT only examine the syntactic structures of candidate segments and cannot sense the semantics of person interactions in those segments. By contrast, our method analyzes the semantics (i.e., interactive verbs) and content (i.e., bigrams) of segments to identify person interactions. Hence, its performance is superior to that of SPT and SDP-CPT. It is noteworthy that SPT and SDP-CPT cannot deal with inter-candidate segments effectively. The reason is that the syntactic structure of inter-candidate segments is usually long and complex, and that affects the methods' detection performance. The proposed method prunes indiscriminative and redundant syntactic constructs in text, so it is effective in detecting inter-candidate segments. SDP-CPT is superior to SPT in terms of intra-candidate segments because the segments are usually short. The corresponding dependency structure is clear that the shortest dependency path of SDP-CPT represents person interactions well. Consequently, it performs better than SPT. FISER also outperforms SPT and SDP-CPT as it incorporates semantic features

to distinguish interactive segments. However, FISER ignores the syntactic structures of text, which are effective in extracting the relations between named entities from text as demonstrated in [21]. It is therefore inferior to our method.

To summarize, the proposed rich interactive tree and bigram kernel approach successfully integrates the syntactic, semantic, and content information in text to identify interactive segments. Hence, it achieves the best precision, recall, and F1 scores among the compared methods, as shown in Table 3.

Table 3. The interaction detection results of the compared methods

System	*Intra-segment*	*Inter-segment*	*Micro-average*	*Macro-average*
	Precision, Recall, F1-score (%)			
SPT	69.42 / 57.10 / 62.66	45.74 / 13.74 / 21.13	63.44 / 36.25 / 46.14	57.58 / 35.42 / 41.90
SDP-CPT	74.34 / 66.86 / 70.40	44.79 / 13.74 / 21.03	67.25 / 41.32 / 51.19	59.57 / 40.30 / 45.72
FISER	80.70 / **81.66** / 81.18	**82.17** / 33.87 / 47.96	81.10 / 58.69 / 68.09	**81.44** / 57.76 / 64.57
Our method	**85.16** / 78.11 / **81.48**	77.27 / **59.74 / 67.36**	**81.70 / 69.28 / 74.98**	81.22 / **68.93 / 74.44**

5 Concluding Remarks

A topic is associated with specific times, places, and persons. Thus, discovering the interactions between the persons would help readers construct the background of the topic and facilitate document comprehension. To this end, we developed a method that combines the rich interactive tree structure and bigram kernel to analyze the syntactic, semantic, and content information in text. It then exploits the derived information to identify interactive segments in topic documents. Our experiment results demonstrate that the proposed method is effective and also outperforms well-known relation extraction and PPI methods.

In the future, we will investigate the syntactic dependency tree and sentimental information in candidate segments to incorporate further syntactic and semantic information into the rich interactive tree structure. We will also utilize information extraction algorithms to extract interaction tuples from interactive segments and construct an interaction network of topic persons.

Acknowledgements. This research was supported by the National Science Council of Taiwan under grant NSC 100-2628-E-002-037-MY3, NSC101-3113-P-001-004, and NSC102-3111-Y-001-012.

References

1. Chen, C.C., Chen, M.C.: TSCAN: A content anatomy approach to temporal topic summarization. IEEE Transactions on Knowledge and Data Engineering 24, 170–183 (2012)
2. Chang, Y.-C., Chuang, P.-H., Chen, C.C., Hsu, W.-L.: FISER: An effective method for detecting interactions between topic persons. In: Hou, Y., Nie, J.-Y., Sun, L., Wang, B., Zhang, P. (eds.) AIRS 2012. LNCS, vol. 7675, pp. 275–285. Springer, Heidelberg (2012)
3. Collins, M., Duffy, N.: Convolution kernels for natural language. In: Proceedings of Annual Conference on Neural Information Processing Systems, pp. 625–632 (2001)

4. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, pp. 423–429 (2004)
5. Feng, A., Allan, J.: Finding and linking incidents in news. In: Proceedings of the 16th ACM International Conference on Information and Knowledge Management, pp. 821–830 (2007)
6. Hong, G.: Relation extraction using support vector machine. In: Proceedings of the 2nd International Joint Conference on Natural Language Processing, pp. 366–377 (2005)
7. Joachims, T.: Text categorization with support vector machine: learning withmany relevant features. In: Proceedings of 10th European Conference on Machine Learning, pp. 137–142 (1998)
8. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions, pp. 178–181 (2004)
9. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing, 1st edn. MIT Press, Cambridge (1999)
10. Miwa, M., Thompson, P., Ananiadou, S.: Boosting automatic event extraction from the literature using domain adaptation and coreference resolution. Bioinformatics 28(13), 1759–1766 (2012)
11. Miyao, Y., Sagae, K., Satre, R., Matsuzaki, T., Tsujii, J.: Evaluating contributions ofnatural language parsers to protein-protein interaction extraction. Bioinformatics 25(3), 394–400 (2009)
12. Moschitti, A.: A study on convolution kernels for shallow semantic parsing. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, pp. 21–26 (2004)
13. Nallapati, R., Feng, A., Peng, F., Allan, J.: Event threading within news topics. In: Proceedings of the 13th ACM International Conference on Information and Knowledge Management, pp. 446–453 (2004)
14. Ono, T., Hishigaki, H., Tanigam, A., Takagi, T.: Automated extraction of informationon protein-protein interactions from the biological literature. Bioinformatics 17(2), 155–161 (2001)
15. Qian, L.H., Zhou, G.D.: Tree kernel-based protein–protein interaction extraction from biomedical literature. Journal of Biomatical Informatics 45(3), 535–543 (2012)
16. Qian, L.H., Zhou, G.D., Zhu, Q.M., Qian, P.D.: Exploiting constituent dependencies fortree kernel-based semantic relation extraction. In: Proceedings of 22nd International Conference on Computational Linguistics, pp. 697–704 (2008)
17. Vernon, G.M.: Human interaction: An introduction to sociology, 1st edn. Ronald Press Co., New York (1965)
18. Xiao, J., Su, J., Zhou, G.D., Tan, C.L.: Protein-protein interaction extraction: a supervisedlearning approach. In: Proceedings of the 1st International Symposium on Semantic Mining in Biomedicine, pp. 51–59 (2005)
19. Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. The Journal of Machine Learning Research 3, 1083–1106 (2003)
20. Zhou, G.D., Qian, L.H., Fan, J.X.: Tree kernel-based semantic relation extraction with rich syntactic and semantic information. Journal of Information Science 180(8), 1313–1325 (2010)
21. Zhang, M., Zhang, J., Su, J., Zhou, G.D.: A composite kernel to extract relations between entities with both flat and structured features. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pp. 825–832 (2006)