Constructing Topic Person Interaction Networks Using a Tree Kernel-Based Method

Yung-Chun Chang, Zhong-Yong Chen, Chien-Chin Chen, and Wen-Lian Hsu

***Abstract*—In this paper, we investigate the interactions between topic persons to help readers construct the background knowledge of a topic. We proposed a rich interactive tree structure to represent syntactic, content, and semantic information in the text for extracting person interactions. Subsequently, a model-based EM method is employed to discover the stance communities of the topic persons to assist the exhibition of the interaction networks. Empirical evaluations demonstrate that the proposed method is effective in detecting and extracting the interactions between topic persons in the text, and outperforms other extraction approaches used for comparison. Furthermore, readers will be able to easily navigate through the topic persons of interest within the interaction networks, and further construct the background knowledge of the topic to facilitate comprehension.**

***Index Terms*—Topic summarization, interaction extraction, person multi-polarization, stance community identification, topic person interaction network.**

I. Introduction

The web has become a powerful medium for disseminating information about diverse topics, such as political issues and sports tournaments. While the web is a rich source of topic information, enormous topic documents often overwhelm topic readers. The problem has motivated the development of many topic mining methods to help readers digest enormous amounts of topic information. For instance, Nallapati *et al.* [1] and Feng and Allan [2] 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 [3] 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 [1]. 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 important persons in a presidential campaign, they can understand documents about the campaign more easily. Interaction discovery is an active research area in the bioinformatics field. A number of studies (e.g., [4], [5]) 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* [6]. 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 investigate the interactions between topic persons and further generate person interaction networks to help readers construct the background knowledge of a topic. Such interactions exemplify different types of human behaviors that make people acknowledge and influence each other. Examples of person interactions include compliments, criticisms, collaborations, and competitions [7]. Recognizing interactions between topic persons is more difficult than analogous tasks of relation extraction (RE) and protein-protein interaction (PPI) extraction, since the interactions between persons are changeable 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. Furthermore, mentions of person interactions may be found across clauses, and may consist of several verbs and named entities that distract their detections and extractions. Thus, topic person interaction mining is a difficult and challenging task.

In light of this, we present a method to generate topic person interaction networks. Due to the massive growth in the number of Chinese documents, Chinese will soon become the second most popular language on the web[1]. Therefore, mining of Chinese interactions has become our topic of interest. Initially, we proposed the *rich interactive tree* (RIT) structure to represent text segments that may convey interactions between person mentions. The RITs of the text segments are applied to classify interactive segments. Next, the interaction extraction then retrieved the interaction keywords of the topic persons from the detected text segments. Finally, we employed the model-based EM method to find the stance communities of the topic persons to assist the exhibition of the interaction networks. Experiment

Manuscript received September 16, 2015; revised December 18, 2015.

Y.-C. Chang, Z.-Y. Chen, and C.-C. Chen are with the Department of Information Management, National Taiwan University, Taiwan (e-mail: changyc@iis.sinica.edu.tw, d98725003@ntu.edu.tw, patonchen@ntu.edu.tw).

W. L. Hsu is with the Institute of Information Science, Academia Sinica, Taiwan (e-mail: hsu@iis.sinica.edu.tw).

[1]International Telecommunication Union (ITU) shows that there is an urgent need for the development of Chinese information processing to explore numerous Chinese information and knowledge.

doi: 10.18178/ijlll.2015.1.4.46

results based on real-world datasets demonstrate that the proposed RIT structure is able to successfully exploit the syntactic structures, interaction semantics, and segment context relevant to person interactions. Consequently, our approach outperforms well-known information extraction methods. Moreover, the constructed topic person interaction networks can help readers construct the background of the topic and facilitate comprehension through a virtualized manner.

II. 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. RE research generally considers relation extraction as a supervised classification task. Given a set of training segments (e.g., a sentence) regarding a specific semantic relation, a supervised classification algorithm is employed to learn a relation classifier. The classifier then determines (classifies) whether a new text segment express the relation or not. To employ a classification algorithm, features are extracted from text segments. Depending on the type of the features, RE methods can be classified as either feature-based or tree kernel-based approaches. The feature-based methods exploit training segments to identify representative text features for relation extraction. For instance, Jiang and Zhai [8] systematically explored the syntactic parse tree and dependency parse tree of text, and elaborated various text features for relation extraction. Their experiments showed that simple text features (e.g., bag-of-word and non-conjunctive entity attribute features) are sufficient to achieve superior relation extraction performance, while over-inclusion of complex features (e.g., adding grammar productions into the syntactic parse tree) might hurt the performance.

The performance of the feature-based methods depends on the selected features. Selecting representative features is challenging and generally requires extensive feature engineering [9]. Although several feature-based methods (e.g., [10], [11]) have explored the parse tree of text, the selected syntactic features hardly comprehend the syntactic structure of text that affect the relation extraction performance. To address this problem, Collins and Duffy [12] 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 [13] first adopted the convolution tree kern (CTK) to resolve the problem of semantic role labeling which can be considered the predecessor of relation extraction. The problem of semantic role labeling is to determine the semantic relationship between a predicate and an argument in a given text segment. The convolution tree kernels examines the syntactic parse tree containing the predicate and the argument, and assigns the text segment a semantic relationship if the segment is syntactically similar to the segments of that relationship in a training corpus. Instead of examining the entire parse tree, Zhang *et al.* [14] incorporated the shortest path-enclosed tree (SPT) which is the sub-tree enclosed by the shortest path linking two entities in a parse tree into CTK for relation extraction. Their experiments demonstrate that SPT expresses the syntactic relation between entities clearly and the method achieves a superior relation extraction performance on many ACE corpora. Due to the great success, recent RE methods begin to combines CTK with SPT.

Our research is also related to protein-protein interaction (PPI) detection [4] that focuses on recognizing protein interactions mentioned in biomedical literature. In biomedical 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. The methods extract lexical, syntactic, and semantic features from text to construct classification models which distinguish text segments that specify protein interactions. For instance, Ono *et al.* [5] 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 to recognize the protein-protein interaction described in a sentence. Nevertheless, these features hardly represent structured and syntactic dependency of text, which are essential for protein-protein interaction detection. In light of this, many tree kernel-based PPI detection methods have been developed. For example, Miyao *et al.* [15] thoroughly compared various parse tree representations on PPI detection. Their comparison result indicates that every parse tree representation has its own merits, so using a single parse tree representation is insufficient for PPI detection. The authors further demonstrated that by combining the dependency parsing and the syntactic deep parse tree their PPI method achieves the best performance.

Our research differs from RE and PPI detection because they aim to determine static and permanent relations between entities. In contrast, our research studies person interactions which are diverse and changeable. To capture the sophisticated nature of person interactions, we integrated the syntactic, content, and semantic information of a text into a rich interactive tree structure to discriminate person interactions from text.

III. Recognizing Topic Person Interactions

Our method first decomposes the topic documents into a set of *candidate segments*, each of which is likely to mention interactions of topic persons. As the syntactic information of text (e.g., parse tree) has proven to be useful in resolving the relationship between entities [10], [14], [15], 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 content of the segment is examined to ornament the rich interactive tree with interactive semantics. We adopted the convolution tree kernel [12] to measure the similarity between text segments in terms of their RITs. The tree kernel is incorporated into the support vector machine (SVM) [16] to learn a classifier for each structural type,

which detect and classifies interactive segments in the topic documents. Subsequently, the interaction tuple extraction identifies keywords of person interactions from the detected interactive segments.

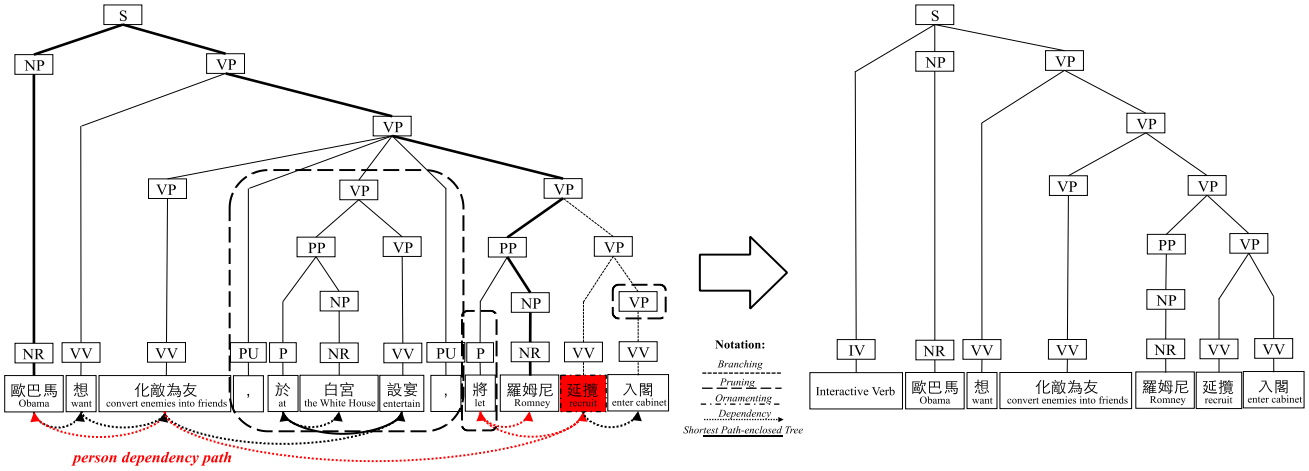

Fig. 1. An example of rich interactive tree construction.

A. Candidate Segment Generation

Fig. 2 shows our candidate segment generation algorithm. For a Chinese topic document d, we first decompose the document into a sequence of clauses $C = \{c_1,..., c_k\}$. Then a Chinese named entity recognition tool is employed to label the tokens in the clauses that represent a person's name. We observed that the rank-frequency distribution of the labeled person names followed the Zipf's law [9], meaning that many of them rarely occurred in the topic documents. Mentions with low frequencies 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 frequent topic person names, referred to as *target persons* hereafter. For any target person pair (p_i, p_j) in P, the candidate segment generation component extracts text segments that are likely to mention their interactions from the document. The component processes the clauses in C individually and considers a clause as the initial clause of a candidate segment if it contains target person $p_i(p_j)$. Since the interaction between p_i and p_j may be narrated by a sequence of clauses, we consider two types of candidate segments, namely *intra-candidate segments* and *inter-candidate segments*. The component then examines the initial clause and subsequent clauses until it reaches an end clause that contains the target person $p_j(p_i)$. If the initial clause is identical to the end clause, the process generates an intra-candidate segment; otherwise, it generates an inter-candidate segment. Note that if there is a period between the clauses of the inter-candidate segment, we drop the segment because p_i and p_j belong to different discourses. In addition, if $p_i(p_j)$ appears more than once in an inter-candidate segment, we truncate all the clauses before the last $p_i(p_j)$ to make the candidate segment concise. By running all target person pairs of P over the topic documents, we can obtain a candidate segment set $CS = \{cs_1,\ldots, cs_m\}$.

B. Rich Interactive Tree Construction

A candidate segment is represented by the rich interactive tree (RIT) structure. Fig. 1 illustrates the process of generating a RIT. By default, we utilize the shortest path-enclosed tree (SPT) as our RIT sapling, because [14] shows that the SPT is effective in identifying the relations between two entities mentioned in a segment of text. The SPT is the smallest sub-tree of the segment's syntactic parsing tree that links person names p_i and p_j. However, the interaction expression is excluded from the SPT if it follows p_j. To remedy this problem, 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.

Candidate Segment Generation
INPUT: $D = \{d_1,\ldots, d_t\}$ – a set of topic documents; $P=\{p_1,\ldots, p_e\}$ – topic persons.
BEGIN
$CS = \{\}$ – candidate segment set of topic
  FOR EACH TOPIC DOCUMENT d_r
    FOR EACH TOPIC PERSON PAIR (p_i, p_j) **in** P
    $C = \{c_1,\ldots, c_k\}$ – a sequence of clauses from d_r.
    inCandidate = false
      FOR $l = 1$ **TO** $l = k$
        IF c_l contains p_i (p_j) && *inCandidate == false*
          add c_l into cs
      inCandidate = true
        ELSE IF c_l contains p_i (p_j) && *inCandidate == true*
          $cs = \{\}$
      add c_l into cs
        ELSE IF c_l contains p_j (p_i) && *inCandidate == true*
      add c_l into cs
      save cs into candidate segment set CS
      inCandidate = false
      $cs = \{\}$
        ELSE IF *inCandidate == true* && c_l has a period
      $cs = \{\}$
       inCandidate = false
        END FOR
    END FOR EACH TOPIC PERSON PAIR
  END FOR EACH TOPIC DOCUMENT
RETURN ***CS***
END

Fig. 2. Candidate segment generation algorithm.

To make the RIT concise and clear, we prune redundant elements in the RIT. We start by truncating inter-candidate segments, because middle clauses of inter-candidate segments are sometimes irrelevant to person interactions. To discriminate middle clauses associated with the topic persons, we adopted the Stanford parser [17], which labels dependencies between text tokens (words). 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. We search for the *person dependency path*,

which we defined as the shortest connecting path of the topic persons in *G*. Then, the pruning operator removes a middle clause and all of its elements in RIT if the clause is not involved in the person dependency path. The clause is pruned because it is not associated with the topic persons. Additionally, since frequent words are not useful in expressing interactions between topic persons, we remove indiscriminative RIT elements. A well-known Chinese stop word list is compiled by collecting the most frequent words in the Sinica corpus[2]. When a word in RIT matches the list, it is removed with its corresponding elements. Finally, duplicate RIT elements are merged, since nodes in an RIT are sometimes identical to their parents. The tree-based kernel used to classify a candidate segment computes the overlap between the RIT structure of the segment and that of the training segments. Considering that complex RIT structures degrade the computation of the overlap, we merge all duplicate elements to make the RIT concise.

Verbs are often good indicators of interactive segments, but not all of them express person interactions. Highlighting verbs (referred to as *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 person dependency path 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.

C. Convolution Tree 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 leverage the convolution tree kernel to capture the syntactic similarity between rich interactive trees. A convolution kernel aims to capture structured information in terms of substructures. Generally, we can represent a parse tree T by a vector of integer counts of each sub-tree type (regardless of its ancestors):

$$\phi(T) = (\#\text{subtree}_1(T), ..., \#\text{subtree}_i(T), ..., \#\text{subtree}_n(T)) \quad (1)$$

where $\#\text{subtree}_i(T)$ is the occurrence number of the i^{th} sub-tree type (subtree_i) in T. Since the number of different sub-trees is exponential with the parse tree size, it is computationally infeasible to directly use the feature vector $\psi(T)$. To solve this computational issue, we leverage the convolution tree kernel [12] to capture the syntactic similarity between the above high dimensional vectors implicitly. 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:

$$K_{CTK}(RIT_1, RIT_2) = \sum_{n_1 \in N_1, n_2 \in N_2} \Delta(n_1, n_2) \quad (2)$$

where N_1 and N_2 are the sets of nodes in RIT_1 and RIT_2, 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 n_1 and n_2 are pre-terminals (POS tags), $\Delta(n_1, n_2)=1\times\lambda$;
3) Else calculate $\Delta(n_1, n_2)$ recursively as:

$$\Delta(n_1, n_2) = \lambda \prod_{k=1}^{\#ch(n_1)} (1 + \Delta(ch(n_1, k), ch(n_2, k))) \quad (3)$$

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. The time complexity of this kernel is $O(|N_1| \bullet |N_2|)$.

D. Interaction Tuple Extraction

Once an interactive segment is detected, we examine its RIT to extract the keyword that depicts the interaction of the topic persons. The extracted keyword and the persons form a tuple $t = (p_i, p_j, \text{token}_{\text{interactive}})$, where p_j and p_i are the topic persons of the segment, and $\text{token}_{\text{interactive}}$ denote the extracted interaction keyword. If a verb of the person dependency path is found in the compiled interactive verb list, the RIT is decorated by attaching an IV tag. Hence, for an interactive segment with an IV tag, we consider the verb as the interaction keyword. It is worth noting that the RIT we proposed comprises not only interactive semantics, but also the syntactic structure of interactive segments. Consequently, a segment with no IV tag may still be classified as interactive as long as its syntactic structure is frequently used to describe person interactions. In Mandarin Chinese, the predicate structure, which usually surrounds the object, is used to determine the topic-comment structure of a clause [15], [18]. Based on this rationale, our system extracts interaction keywords from an interactive segment with no IV tag using the following rule: If there are existing verbs between the topic persons, we label the verb before the last topic person in the segment's RIT as the interaction keyword. Otherwise, we label the last verb of the RIT as the interaction keyword.

IV. Stance Community Identification of Topic Persons Using Model-Based EM Method

To exhibit the interaction network, we leverage a model-based EM method to determine the stance communities of the topic persons [19]. First of all, the EM method employed the Stanford named entity recognizer (hereafter referred to as NER) to extract person names from the topic documents. Then, it clusters the persons into the stance communities, each of which has the same goal or reaches a consensus. More specifically, given a set of topic documents $D= \{d_1, d_2, \ldots, d_N\}$, the model-based EM method extracted person names $P = \{p_1, p_2, \ldots, p_M\}$ from the set of

[2]http://www.aclclp.org.tw/use_wlawf.php

documents D and constructed person vectors p_i whose entries $p_{i,j}$ indicate the frequency of the person p_i in the document d_j as shown in Fig. 3.

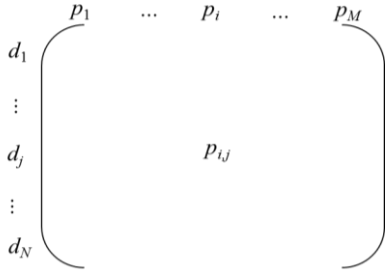

Fig. 3. Person-document matrix *DP*.

Assume the documents contain K stance communities. Let $\theta = \{(\alpha_1, \omega_1), (\alpha_2, \omega_2), \ldots, (\alpha_k, \omega_k)\}$ be the parameters of the stance community models, in which α_k represents the weight of community k and ω_k represents the representative vector of community k. Thus, the EM-method models the stance community task as follows.

$$\hat{\theta} = \arg\max_{\theta \in \text{search space}} P(\theta | P) \tag{4}$$

Given the set of person P, equation (4) searches for the appropriate model parameters. Subsequently, it can analyze and solve it by the Bayes theorem [9] as shown below. As the number of stance community models is infinite, we assume that all models have the same prior probability.

$$\begin{aligned} \hat{\theta} &= \arg\max_{\theta \in \text{search space}} P(\theta) \times P(P | \theta) \\ &= \arg\max_{\theta \in \text{search space}} P(P | \theta) \\ &= \arg\max_{\theta \in \text{search space}} \prod_{i=1}^{M} \sum_{k=1}^{K} \alpha_k \times P(\underline{p}_i | \underline{\omega}_k) \end{aligned} \tag{5}$$

As the posterior probability was converted into the likelihood function, the design of this function is essential for the task. Kanayama and Nasukawa demonstrated [20] that the text units tend to co-occur in the same context to make the topic documents coherent. As a result, the likelihood function was learned from the topic documents D by using the standard correlation coefficient that measures joint behaviors of the persons and the representative vector of each community. Data sparseness has always been an issue in the text mining field, and Chen and Chen [19] employed the weighted correlation coefficient below to manage this problem.

$$P(\underline{p}_i | \underline{\omega}_k) = \frac{wcor\wedge(\underline{p}_i, \underline{\omega}_k)}{\sum_{i=1}^{M} wcor\wedge(\underline{p}_i, \underline{\omega}_k)}, \tag{6}$$

where

$$wcor\wedge(\underline{p}_i, \underline{\omega}_k) = \frac{S_{xy}}{S_x * S_y}, \tag{7}$$

$$\begin{aligned} S_{xy} = (1-\beta) \sum_{d \in co(i,k)} (p_{i,d} - \bar{p}_i) * (\omega_{k,d} - \bar{\omega}_k) + \\ \beta \sum_{d \in D \backslash co(i,k)} (p_{i,d} - \bar{p}_i) * (\omega_{k,d} - \bar{\omega}_k) \end{aligned} \tag{8}$$

$$S_x = \sqrt{(1-\beta) \sum_{d \in co(i,k)} (p_{i,d} - \bar{p}_i)^2 + \beta \sum_{d \in D \backslash co(i,k)} (p_{i,d} - \bar{p}_i)^2} \tag{9}$$

$$S_y = \sqrt{(1-\beta) \sum_{d \in co(i,k)} (\omega_{k,d} - \bar{\omega}_k)^2 + \beta \sum_{d \in D \backslash co(i,k)} (\omega_{k,d} - \bar{\omega}_k)^2}. \tag{10}$$

Equation (6) represents the probability of person p_i belonging to the stance community k, and equation (7) is the weighted correlation coefficient. The numerator of equation (7) indicates the co-variance of the person vectors and the representative vectors of the communities, and the denominators S_x and S_y represent the standard deviation of the person vectors and representative vectors, respectively. β is a weight to adjust the importance of the non-co-occurrence of person p_i and community k. The set $co(i, k)$ indicates the set of documents that person p_i and ω_k co-occur. $\bar{p}_i$ and $\bar{\omega}_k$ are the average frequencies of the person vector p_i and the representative vector ω_k.

Initially, the EM-method randomly selected the person vectors as the representative vectors. Afterwards, the method employed the EM steps to calculate the probability of which community the topic person belongs, and re-computes the weight and the representative vector of each stance community. The EM steps are as follows.

E-step:

$$E[z_{i,k}] = \frac{\alpha_k * P(\underline{p}_i | \underline{\omega}_k)}{\sum_{j=1}^{K} \alpha_j * P(\underline{p}_i | \underline{\omega}_j)}, \tag{11}$$

M-step:

$$\begin{aligned} \alpha_k &= \sum_{i=1}^{M} E[z_{i,k}] / M \text{ and} \\ \underline{\omega}_k &= \sum_{i=1}^{M} E[z_{i,k}] * \underline{p}_i / \sum_{i=1}^{M} E[z_{i,k}]. \end{aligned} \tag{12}$$

$E[z_{i,k}]$ represents the expected value of person i belonging to stance community k. The EM steps are performed recursively until convergence, and then the EM-method assigns person i to the community with the maximum expected value. The process of identifying the stance community is shown in Fig. 4.

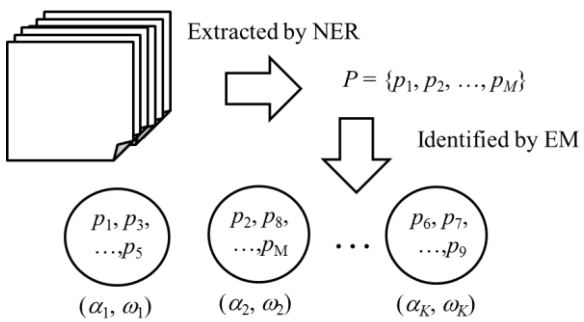

Fig. 4. The procedure of identifying stance communities by EM.

V. Experiments

In this section, we introduce the evaluation dataset and metrics. Moreover, we compared the performance of the proposed method with other well-known relation extraction and PPI methods, and demonstrate the topic person interaction networks.

A. Dataset and Setting

To the best of our knowledge, there is no official corpus for person interaction detection. Therefore, we compiled a data corpus to evaluate the performance of the proposed method. The data corpus comprises 24 important Chinese topics from 2004 to 2014. Each topic consists of 100 Chinese news documents collected from Yahoo News. We employed the Stanford NER [21] to tag person names mentioned in the topic documents, which resulted in a total of 15,370 person names that represent 665 unique persons. To reduce the effect of irrelevant person names on system performance, we evaluated the most frequent person names whose frequency reached 70% of the total person names in the topic documents. All person names that meet this criterion represent important topic persons. The candidate segment generation algorithm extracts 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, indicating that the data corpus is reliable. As shown in Table I, a great portion (i.e., 47.5%) of interactive segments are inter-clausal, while around half (49.8%) of the intra-clausal segments are non-interactive. In other words, persons that exist in the same clause generally have no interactions, and the interaction of persons is usually narrated by a sequence of clauses. The distributions reveal that the mining of person interactions is not trivial.

TABLE I: Statistics of the Data Corpus

# of topics	24
# of topic documents	2400
# of tagged person names	665
# of evaluated person names	140
# of candidate segments	9632
# of interactive segments (intra)	2107
# of interactive segments (inter)	1912
# of non-interactive segments (intra)	2092
# of non-interactive segments (inter)	3521
# of interaction keywords	4019

With respect to the convolution kernels of the RIT, Moschitti's tree kernel toolkit [13] was adopted and implemented into the SVMLight package [16] for RIT classification. To derive credible evaluation results, we utilized the leave-one-out cross validation approach [9]. Specifically, we evaluate SPIRIT for multiple runs. For each run, the candidate segments of a topic is selected for testing, and the remaining topics are used for the RIT classifier training. The testing results over all topics then are averaged to obtain the global system performance. The evaluation metrics are the precision rate, recall rate, and F_1-measure [9]. The F_1 measure is used to determine the relative effectiveness of the compared methods.

B. Results

For experiments on interaction detection and extraction, we compared our method with three extraction systems: the Chinese open information extraction approach [22] (denoted as *CORE*), the CRF-based relation descriptors extraction method [23] (denoted as *CRF-RDE*), and the SVM-based approach for biomedical trigger extraction [24] (denoted as *SVM-BTE*). Note that the compared systems are designed to label relation keywords in a given text, similar to the purpose of identifying interactive keywords of a candidate segment for its classification. We first examined the performance of all systems on interaction detection. As shown in Table II, the precision rate of CORE is inferior to the proposed method. This is because CORE regards existing verbal phrases in the parse tree of a segment as a cue of person interactions. In the evaluation dataset, a considerable number of non-interactive segments contain verbal phrases. As a result, many false positives were generated by CORE to harm its precision rate. SVM-BTE achieved the highest recall of all since it explores the bigrams and trigrams of interactive verbs to detect interactive segments. Considering that Chinese keywords are generally compounds [25], [26], many of the bigrams and trigrams also convey interaction semantics. Consequently, the method is able to capture a lot of interactive segments to increase the recall rate. Nevertheless, SVM-BTE overlooks the person dependency path, so many of the bigrams and trigrams that exist in the detected segments are unrelated to the topic persons. Therefore, many false positives were generated to deteriorate its detection precision. Notably, the CRF-RDE obtained the highest precision as it employed stringent feature functions which examines the context of entities in a given text, resulting in high accuracy when detecting interactive segments. 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. However, these feature functions are too rigid and lead to a low recall rate which deteriorates the system performance. By contrast, the proposed method not only prunes indiscriminative and redundant syntactic constructs in text, but also examines the content of the segment to ornament the rich interactive tree with interactive semantics, thus outperforming other systems.

TABLE II: Performance Evaluation of Different Systems

System	Detection	Extraction
	Precision / Recall / F_1-measure (%)	
CORE	46.53 / 46.75 / 46.64	30.15 / 23.19 / 26.22
CRF-RDE	**79.00** / 30.33 / 43.83	**77.88** / 28.39 / 41.61
SVM-BTE	43.64 / **77.81** / 55.92	22.58 / 29.30 / 25.50
Our method	67.66 / 59.60 / **63.38**	59.56 / **41.15** / **48.68**

To further investigate the competence of our system, interaction extraction performance of all four methods were also evaluated. The CORE and SVM-BTE achieved the lowest performance among all methods with an average P/R/F around 26%. As CORE is a grammar-based extraction approach, it recognizes verbal phrases of a clause to extract interactions. Nonetheless, a great deal of interaction keywords are not included in the verb phrase, so the coverage and accuracy of CORE is limited. The feature functions of SVM-BTE were inefficient in extracting interactions and produce many false positives to worsen the overall performance. Since the CRF-RDE utilizes stringent feature functions which examines the context of entities in a given

text, the extraction performance of CRF-RDE is similar to its detection with slightly reduced precision and recall. On top of that, the proposed method is able to filter non-interactive segments efficiently, reducing the chance of making false positive errors to achieve the best overall performance.

C. Topic Person Interaction Network

After extracting interaction tuples and identifying the stance communities of the topic persons, we can then generate topic person interaction networks to link all the extracted information together. Unlike networks constructed from binary interactions, a topic person interaction network further defines the descriptors of the interactions. Fig. 5 illustrates a topic person interaction network constructed from the topic of the 2012 U.S. presidential election by the proposed method. As shown in this figure, the interactions between Barack Obama and Mitt Romney are generally negative (e.g.批評 (criticize) and 諷刺 (satirize)). We can also observe that Joe Biden and Paul Ryan have similar interactions. It is intriguing that a positive interaction of them "共進午餐 (have lunch)" is extracted between Barack Obama and Mitt Romney, which is contrary to our expectations. We speculate that this interaction is topic specific, since it is derived from the text focusing on President Obama forming a new cabinet. This indicates that person interactions are changeable and topic-dependent. On the other hand, the EM model-based method can correctly identify the stance of the topic persons. We observed that the persons with the same stance frequently co-occur in most of the documents because journalists tend to analyze and interview the persons of the same camp in the same document. Since the method identifies the stances in terms of the word usage of the person names in the topic documents, it can thus correctly identify the stance communities. With the topic person interaction networks, reader will be able to easily navigate through their topic persons of interest, and further construct the background of the topic to facilitate comprehension.

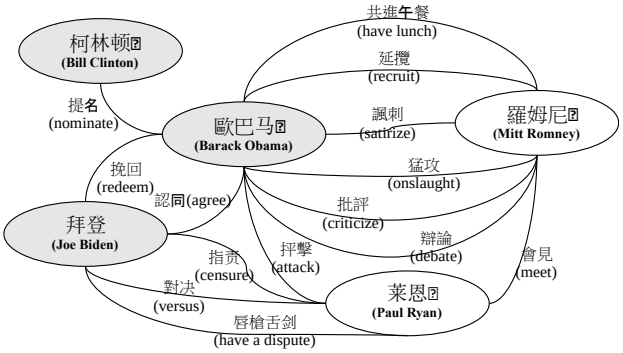

Fig. 5. Generated topic person interaction network related to the 2012 U.S. presidential election.

VI. Concluding Remarks

A topic is associated with specific times, places, and persons. Thus, discovering the interactions between persons would help readers gain increased understanding of the topic. In this paper, we proposed a rich interactive tree structure for recognizing interaction between topic persons, and leveraged a model-based EM method to identify the stance community of the topic persons for generating topic person interaction networks. Experiment results based on real-world datasets demonstrate that the proposed RIT structure is able to successfully exploit the syntactic structures, interaction semantics, and segment context relevant to person interactions. Hence, our method outperforms other extraction methods. Furthermore, readers can easily navigate through the topic persons of interest within the interaction networks, and further construct the background knowledge of the topic to facilitate comprehension.

In the future, we will investigate the sentimental information in candidate segments to incorporate supplementary syntactic and semantic information into the rich interactive tree structure. Furthermore, we would like to merge multiple similar interactions and unify them as one synthesized interaction.

Acknowledgment

We are grateful to the anonymous reviewers for insightful comments. This research was supported by the Ministry of Science and Technology of Taiwan under grant MOST 103-2221-E-002-106-MY2 and 103-3111-Y-001-027.

References

[1] R. Nallapati, A. Feng, F. Peng, and J. Allan, "Event threading within news topics," in *Proc. the 13th ACM International Conference on Information and Knowledge Management*, 2004, pp. 446-453.

[2] A. Feng and J. Allan, "Finding and linking incidents in news," in *Proc. the 16th ACM International Conference on Information and Knowledge Management*, 2007, pp. 821-830.

[3] C. C. Chen and M. C. Chen, "TSCAN: A content anatomy approach to temporal topic summarization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, pp. 170-183, 2012.

[4] J. Xiao, J. Su, G. D. Zhou, and C. L. Tan, "Protein-protein interaction extraction: A supervised learning approach," in *Proc. the 1st International Symposium on Semantic Mining in Biomedicine*, 2005, pp. 51-59.

[5] T. Ono, H. Hishigaki, A. Tanigam, and T. Takagi, "Automated extraction of information on protein-protein interactions from the biological literature," *Bioinformatics*, vol. 17, no. 2, pp. 155-161, 2001.

[6] M. Miwa, P. Thompson, and S. Ananiadou, "Boosting automatic event extraction from the literature using domain adaptation and coreference resolution," *Bioinformatics*, vol. 28, no. 13, pp. 1759-1766, 2012.

[7] G. M. Vernon, *Human Interaction: An Introduction to Sociology*, 1st Ed. New York: Ronald Press Co., 1965.

[8] J. Jiang and C. Zhai, "A systematic exploration of the feature space for relation extraction," in *Proc. Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics*, 2007, pp. 113-120.

[9] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[10] G. D. Zhou, J. Su, J. Zhang, and M. Zhang, "Exploring various knowledge in relation extraction," in *Proc. the 43th Annual Meeting of the Association for Computational Linguistics*, 2005, pp. 427-434.

[11] N. Kambhatla, "Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations," in *Proc. the 42nd Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions*, 2004, pp. 178-181.

[12] M. Collins and N. Duffy, "Convolution kernels for natural language," in *Proc. Annual Conference on Neural Information Processing Systems*, 2001, pp. 625-632.

[13] A. Moschitti, "A study on convolution kernels for shallow semantic parsing," in *Proc. the 42nd Annual Meeting of the Association for Computational Linguistics*, 2004, pp. 21-26.

[14] M. Zhang, J. Zhang, J. Su, and G. D. Zhou, "A composite kernel to extract relations between entities with both flat and structured features," in *Proc. the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, 2006, pp. 825-832.

[15] Y. Miyao, K. Sagae, R. Satre, T. Matsuzaki, and J. Tsujii, "Evaluating contributions of natural language parsers to protein-protein interaction extraction," *Bioinformatics*, vol. 25, no. 3, pp. 394-400, 2009.

[16] T. Joachims, "Text categorization with support vector machine: learning with many relevant features," in *Proc. 10th European Conference on Machine Learning*, 1998, pp. 137-142.
[17] J. R. Finkel, T. Grenager, and C. D. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proc. the 43nd Annual Meeting of the Association for Computational Linguistics*, 2005, pp. 363-370.
[18] K. V. Prince, "Predication and information structure in mandarin Chinese," *Journal of East Asian Linguistics*, vol. 21, no. 4, pp. 329-366, 2012.
[19] C. C. Chen and Z. Y. Chen, "A model-based EM method for topic person name multi-polarization," in *Proc. the 7th Asia conference on Information Retrieval Technology*, 2011, pp. 410-421.
[20] H. Kanayama and T. Nasukawa, "Fully automatic lexicon expansion for domain-oriented sentiment analysis," in *Proc. the 2006 Conference on Empirical Methods in Natural Language Processing*, 2006, pp. 355-363.
[21] R. Levy and C. D. Manning, "Is it harder to parse Chinese, or the Chinese treebank?" in *Proc. the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 439-446, 2003.
[22] Y. H. Tseng, L. H. Lee, S. Y. Lin, B. S. Liao, M. J. Liu, H. H. Chen, O. Etzioni, and A. Fader, "Chinese open relation extraction for knowledge acquisition," in *Proc. the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 12-16.
[23] Y. Li, J. Jiang, H. L. Chieu, and K. M. Chai, "Extracting relation descriptors with conditional random fields," in *Proc. the 5th International Joint Conference on Natural Language Processing*, 2011, pp. 392-400.
[24] J. Björne and T. Salakoski, "Generalizing biomedical event extraction," in *Proc. the BioNLP Shared Task 2011 Workshop*, 2011, pp. 183-191.
[25] C. N. Li and S. A. Thompson, "Mandarin Chinese: A functional reference grammar," Crane, Taipei, 1981.
[26] Y. R. Chao, *A Grammar of Spoken Chinese*, Berkeley: University of California Press, 1968.

Yung-Chun Chang is a PhD student at National Taiwan University in Taiwan. He received his M.S. degree in information management from Chang Jung Christian University, Taiwan in 2007. His research interests include natural language, knowledge discovery from text, information retrieval, and relation extraction. His research interests in developing effective methodologies for extracting valuable knowledge from the world wide web. His recent works included: topic-oriented social relationship extraction, principle-based approach for NLP applications, text mining for sentiment analysis, temporal relation extraction from patient discharge summaries, and text categorization.

Zhong-Yong Chen received the MS degree in information management from the National Kaohsiung University of Applied Sciences, Taiwan, in 2009. He is currently working toward the PhD degree in information management at the National Taiwan University, Taiwan. His research interests include information retrieval and text mining.

Chien-Chin Chen received the 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 the 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. He was a tenured associate professor in Northwestern University before joining the Institute of Information Science in Academia Sinica as a research fellow in 1989. Dr. Hsu's 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 a Chinese input software, GOING, which has since revolutionized Chinese input on computer. Dr. Hsu is particularly interested in applying natural language processing techniques to understanding DNA sequences as well as protein sequences, structures and functions and also to biological literature mining. Dr. Hsu received the Outstanding Research Award from the National Science Council 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. He was the president of the Artificial Intelligence Society in Taiwan from 2001 to 2002 and the president of the Computational Linguistic Society of Taiwan from 2011 to 2012.