Survey on :Multi Attribute Behavior Analysis Model in Collaborative Environment

P.Girija [1], Dr.V.Jeyabalaraja [2]
[1, 2] Department of CSE
[1] M.E, Velammal Engineering College.
[2] Professor, Velammal Engineering College.

Abstract- *A topic person interaction detection method called SPIRIT, which classifies the text segments in a set of topic documents that convey person interactions to represent syntactic, context, and semantic information of text, and this structure is incorporated into a tree-based convolution kernel. 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. Nature of the person is represented through behavior and mining technique helps to analyze the opinion a person exhibits. Discovering semantic knowledge is significant for understanding and interpreting how people interact in a meeting discussion. Patterns of human interaction is extracted from the minutes of the meetings. Different Human interactions, such as proposing an idea, giving comments, and acknowledgements, indicate user intention toward a topic or role in a discussion. To further understand and interpret human interactions in meetings, we need to discover higher level semantic knowledge about them, such as which interaction often occur in a discussion, what interaction flow a discussion usually follow, and what relationship exist among interactions. This knowledge describe important patterns of interaction. Based on the human interaction the behavior of the members are identified and people of similar nature are grouped together.*

Keywords- Topic person, Sentiment mining, topic analysis, co-extracting relation.

I. INTRODUCTION

The Web has become an abundant source of information because of the prevalence of Web2.0, and Internet users can express their opinions about topics easily through various collaborative tools, such as weblogs. Published documents provide a comprehensive view of a topic, but readers are often overwhelmed by large number of topic documents. To help readers comprehend numerous topic documents, several topic mining methods have been proposed. Information extraction is an important research topic in natural language processing. It tries to find relevant information from the large amount of text documents available in digital archives and on the World Wide Web. Research on information extraction has been promoted by the Message Understanding Conferences (1987–1998) and the Automatic Content Extraction program [1]. According to the ACE program, information extraction subsumes a broad range of tasks, including entity detection and tracking, Relation Detection and Characterization (RDC), and event detection and characterization. This paper focuses on the extraction of semantic relations between named entities, as defined by the ACE RDC task, which detects and classifies semantic relationships (usually of predefined types) between pairs of entities. According to the ACE program, an entity is an object or a set of objects, while a relation is an explicitly or implicitly stated relationship between two entities.

For example, the sentence ‘‘Bill Gates is the chairman and chief software architect of Microsoft Corporation.” conveys the ACE-style relation ‘‘EMPLOYMENT.exec” between the entities ‘‘Bill Gates” (PER, person) and ‘‘Microsoft Corporation” (ORG, organization). The extraction of semantic relations between entities can be very useful in many applications such as answering questions (like ‘‘Who is the President of the United States?”) and retrieving information,(by expanding the term ‘‘Barack Obama” to ‘‘the President of the United States” via his relationship with ‘‘the United States”).Much research has been performed on the extraction of semantic relations between named entities. Feature vector-based methods [8,10,24–28] recast the semantic relation extraction task as a classification problem first by transforming relation instances into multi-dimensional vectors with various features and then by applying machine learning approaches to detect and classify the semantic relationship between the named entities. These researchers have achieved certain success by employing diverse linguistic features, varying from lexical knowledge and entity-related information to syntactic parse trees.

Sentiment mining refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgment or evaluation, affective state (that is to say, the emotional state of the author when writing), or the intended emotional communication. Opinion mining (sentiment mining, opinion/sentiment extraction) attempts to make the automatic systems to determine the human opinion from text written in natural language. It seeks to identify the view point (s) underlying a text span. Sentiment mining draws on computational linguistic, information retrieval, text mining, natural language processing, machine learning, statistics and predictive analysis. In real life, facts are important, but opinion also plays a crucial role. Search engines do not search for opinions. Opinions are hard to express with a few keywords .

II. RELATED WORKS

Human interaction in meetings has attracted much research in the fields of image/speech processing, computer vision, and human-computer interaction (see [2] for a full review). Stiefelhagen et al. [3] used microphones to detect the current speaker and combined acoustic cues with visual information for tracking the focus of attention in meeting situations. McCowan et al. [5] recognized group actions in meetings by modelling the joint behavior of participants based on a two-layer Hidden Markov Model (HMM) framework. The AMI project [6] was proposed for studying human interaction issues in meetings, such as turn-taking, gaze behavior, influence, and talkativeness. Otsuka et al. [7] used gaze, head gestures, and utterances in determining interactions regarding who responds to whom in multiparty face-to-face conversations. DiMicco et al. [8] presented visualization systems for reviewing a group's interaction dynamics,e.g., speaking time, gaze behavior, turn-taking patterns, and overlapping speech in meetings. In general, the above-mentioned systems aim at detecting and visualizing human interactions in meetings, while our work focuses on discovering higher level knowledge about human interaction. Mining human interactions is important for accessing and understanding meeting content [1]. First, the mining results can be used for indexing meeting semantics, also existing meeting capture systems could use this technique as a smarter indexing tool to search and access particular semantics of the meetings [9], [10]. Second, the extracted patterns are useful for interpreting human interaction in meetings.

Cognitive science researchers could use them as domain knowledge for further analysis of human interaction. Moreover, the discovered patterns can be utilized to evaluate whether a meeting discussion is efficient and to compare two meeting discussions using interaction flow as a key feature. Unlike mining patterns of actions occurring together [11], patterns of trajectories [12], and patterns of activities [13], our study aims at discovering interaction flow patterns in meeting discussions, such as relationships between different types of interactions. We are aiming at identifying human behavior patterns from the interactions. By the identification of the pattern with the human , we can find out the nature of the person during meetings, then the domain of interest and to perform several types of reasoning.

Several works done in discovering human behavior patterns by using stochastic techniques we present SPIRIT, which automatically detects text segments (called interactive segments hereafter)that convey person interactions in a set of topic documents.

III. SYSTEM MODEL

The Symbolic or Knowledge base approach and Machine learning approach are the two strategies used for analyzing sentiments from the text. Symbolic approach requires a large database of predefined emotions and an efficient knowledge representation for identifying sentiments. Machine learning approach uses a training set to develop a sentiment classifier to classify sentiments. 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

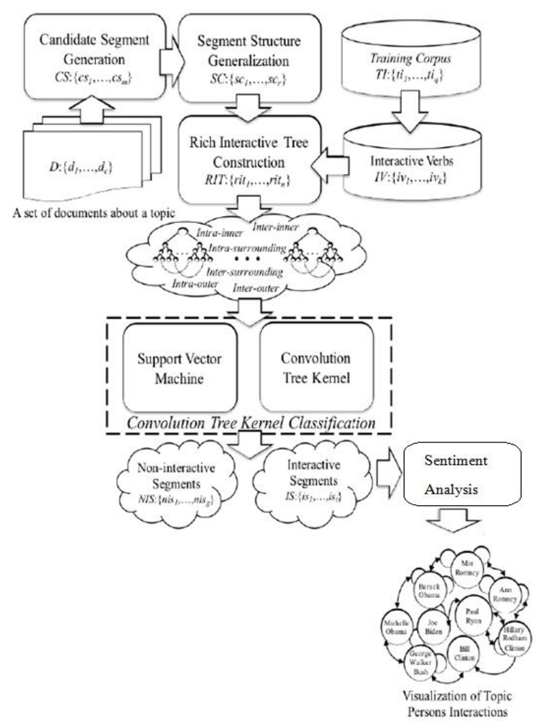


Fig 1 displays the system architecture of SPIRIT, which is comprised of four key components: candidate segment generation, segment structure generalization (SSG), rich interactive tree construction, and convolution tree kernel classification.

text (e.g., parse tree) has proven to be useful in resolving the relationship between entities. 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.

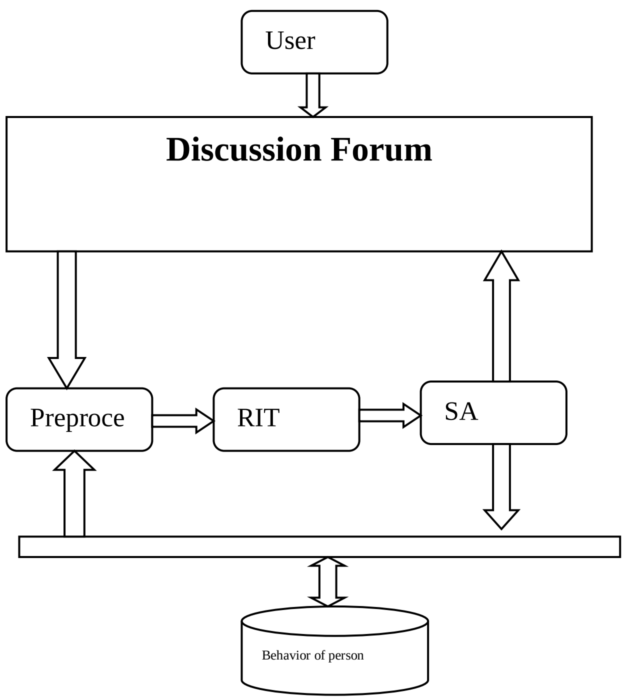


Figure 2: Architecture of Interactive sentiment analysis model

The Figure 1, shows the architecture of the Interactive sentiment analysis model and shows the functional stages in detail.

Preprocessing:

We first decompose the document into a sequence of clauses C = {c1,..., ck}. 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 = {p1,…, pe}denote the set of frequent topic person names, referred to as target persons hereafter. For any target person pair (pi, pj) 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 pi(pj). Since the interaction between pi and pj 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 pj(pi). 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 pi and pj belong to different discourses.

```
Candidate Segment Generation
INPUT: D = {d1,…, di} – a set of topic documents; P={p1,…, pe} –
topic persons.
BEGIN
CS = {} – candidate segment set of topic
    FOR EACH TOPIC DOCUMENT dr
        FOR EACH TOPIC PERSON PAIR (pi, pj) in P
        C = {c1,…, ck} – a sequence of clauses from dr.
        inCandidate = false
            FOR l = 1 TO l = k
                IF cl contains pi (pj) && inCandidate == false
                    add cl into cs
          inCandidate = true
                ELSE IF cl contains pi (pj) && inCandidate == true
                    cs = {}
          add cl into cs
                ELSE IF cl contains pj (pi) && inCandidate == true
           add cl into cs
           save cs into candidate segment set CS
          inCandidate = false
          cs = {}
                ELSE IF inCandidate == true && cl has a period
          cs = {}
           inCandidate = false
            END FOR
        END FOR EACH TOPIC PERSON PAIR
    END FOR EACH TOPIC DOCUMENT
RETURN CS
END
```

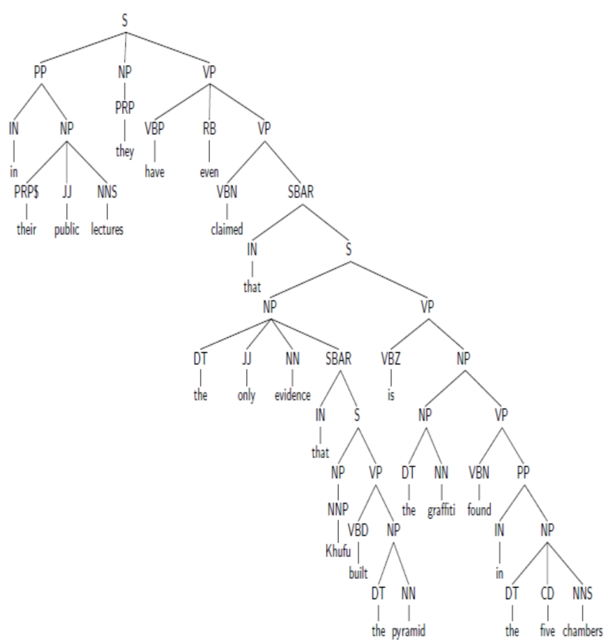
The above discussed algorithm, performs candidate segments.

In addition, if pi(pj) appears more than once in an inter-candidate segment, we truncate all the clauses before the last pi(pj) 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 = {cs1,…, csm}.

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 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 pi and pj. However, the interaction expression is excluded from the SPT if it follows pj. 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.

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 stop word list is compiled by collecting the most frequent words in the Sinica corpus2. 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.

Pseudo Code of MABA:

Input: Preprocessed Trace Pt

Output: Multi Attribute Trust Factor MATF.

Start

For each attribute Ai of request Req

Compute Total number of access.

Tna = $\int_{i=1}^{size(Pt)} \sum Pt(i) \in Ai$

Compute number of completeness.

Nc = $\int_{i=1}^{size(Pt)} \sum (Pt(i) \in Ai) \,\&\&\, Pt(i).status == success$

Compute trust factor of Ai.

Tai = $\frac{Nc}{Tna} \times size(Pt)$

End

Compute Multi attribute trust factor MATF.

MATF = $\int \frac{\sum_{i=1}^{size(Attr)} Tai(Ai)}{size(Attr)}$

Stop

The above discussed algorithm computes the multi attribute trustworthy measure by computing the multi attribute trust factor to decide the trust of any user request.

At this stage, the method uses the above mentioned two modules to perform access control. Upon receiving the request from the user the method performs preprocessing and multi attribute behavioral analysis. Based on the result of multi attribute behavioral analysis the method computes the trust factor to allow or deny the user request.

Sentiment Analysis:

Our study aims at discovering interaction flow patterns in meeting discussions, such as relationships between different types of interactions. We are aiming at identifying human behavior patterns from the interactions. With the identification of the pattern with the human we can find out the nature of the person during meetings. Human Interaction is a vital event to understand communicative information.Understanding human behavior is essential in applications including automated surveillance, video archival/retrieval, medical diagnosis, and human computer interaction. Group social dynamics can be useful for determining whether meeting was well organized and whether the conclusion was rational.

Minutes of meeting are read from the text corpus and preprocessed as given in figure 1. These are then matched with patterns of interactions and are grouped together .They are then classified and form patterns of individual members of the meeting.The words are defined for the features of RIT.

Pseudo code for Pattern matching:

Input: Keywords present in the document
Output: Patterns are formed

1. The keywords are checked with the features defined
2. The matched words are extracted and identified
3. Using Apriori algorithm these words are mined to get a pattern The keywords identified are matched with the lexicon table that has been created for the interactions of Proposal (PRO), Comment (COM) and Acknowledgement (ACK).Using Apriori algorithm similar patterns are mined out. Some examples of patterns can be PRO, COM, ACK, PROCOM, PRO-ACK, PRO-COM-ACK

The features and corresponding persons are identified and placed in a table.

Pseudo code for Grouping:

Input: Matched words from the pattern
Output: A table which contain the patterns of each individual present in the meeting
Step 1: The matched words are counted for the corresponding person
Step 2: They are grouped based on similarity of words
Step 3: Pattern generated for each person based on the data in the table

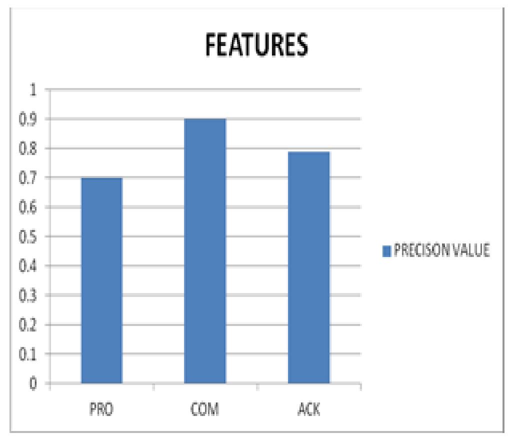
III. RESULTS AND DISCUSSION

Precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also known as sensitivity) is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance. High precision means that an algorithm returned comparatively more relevant results than irrelevant. Precision = true positives/total elements in the positive class i.e. Precision = true positives/ (true positive+ false positives) The three features extracted are PRO, COM, and ACK For the case of PRO – proposal Assert, recommend, inform are identified as Comment and are False Positive. For the case of COM-comment Announce, observe are identified as Acknowledgement and are False Positive. For the case of ACK-acknowledgement Defend, admit are identified as Comment and are False Positive.

Parameter	Value
Tool Used	Advanced Java
Number of resources	100
Number of users	50
Size of trace	100

Table 1: Details of simulation Parameters

The Table 1, shows the details of simulation parameters being used to evaluate the performance of the proposed approach.



The Graph 1, indicates the values obtained when the features Proposal (PRO), Comment (COM), Acknowledgment (ACK) are calculated based on the factors of true positives and false positives. The Precision values when reaches 1 shows maximum accuracy.

IV. CONCLUSION

Based on the interactions among the people present in the meeting we are able to retrieve a pattern for each meeting. Mining results can be used for interpreting human interactions in the meetings. As future work, plan to perform clustering based on the interaction patterns to identify the behavior of each individual in the meeting, thus exploring the involvement of each person in the meeting.. 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.

REFERENCES

[1] A. Feng and J. Allan, “Finding and linking incidents in news,” in Proc. 16th ACM Int. Conf. Inf. Knowl. Manag., 2007, pp. 821–830.

[2] A. Moschitti, "A Study on Convolution Kernels for Shallow Semantic Parsing," Proc. 42nd Annual Meeting of the Association for Computational Linguistics, pp. 21-26, 2004.

[3] C.C.Chen and M.C. Chen, “TSCAN: A content anatomy approach to temporal topic summarization,” IEEE Trans. Knowl.Data Eng., vol. 24, no. 1, pp. 170–183, Jan. 2012.

[4] C.W. Shih, C.W. Lee, R.T. Tsai, and W.L. Hsu, "Validating Contradiction in Texts Using Online Co-Mention Pattern Checking," ACM Transactions on Asian Language Information Processing, vol.11, issue 4, pp. 17:1-12:21, 2012.

[5] D. Croce, A. Moschitti, and R. Basili, "Structured Lexical Similarity via Convolution Kernels on Dependency Trees," Proc. The 2011 Conference on Empirical Methods in Natural Language Processing, pp. 1034-1046, 2011.

[6] D. Croce, R. Basili, A. Moschitti, and M. Palmer, "Verb Classification using Distributional Similarity in Syntactic and Semantic Structures," Proc.the 50th Annual Meeting of the Association for Computational Linguistics,pp.263-272, 2012.

[7] D. Croce, E. Bastianelli, and G. Castellucci, "Structured Kernel-Based Learning for the Frame Labeling over Italian Texts,"Proc.Evaluation of Natural Language and Speech Tools for Italian, pp. 220-229, 2013.

[8] G.D. Zhou, J. Su, J. Zhang, and M. Zhang, "Exploring Various Knowledge in Relation Extraction," Proc. 43th Annual Meeting of the Association for Computational Linguistics, pp. 427-434, 2005.

[9] M. Collins and N. Duffy, "Convolution Kernels for Natural Language," Proc. Annual Conference on Neural Information Processing Systems, pp. 625-632, 2001.

[10] P. Annesi, D. Croce, and R. Basili, "Semantic Compositionality in Tree Kernels," Proc. the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 1029-1038, 2014.

[11] S. Filice, G. Castellucci, D. Croce, and R. Basili, "KeLP: a Kernelbased Learning Platform for Natural Language Processing, " Proc. The 53rd Annual Meeting of the Association for Computational Linguistics-System Demonstrations, pp. 19-24, 2015.

[12] Z. Dmitry, A. Chinatsu, and R. Anthony, "Kernel Methods for Relation Extraction," The Journal of Machine Learning Research, vol.3, 3/1/2003, pp. 1083-1106, 2003.

[13] Yuanbin Wu, Qi Zhang, Xuanjing Huang,Lide Wu, ”Phrase Dependency Parsing for Opinion” , Fudan University School of Computer Science , ACL 2009.

[14] Ivan Titov and ryan McDonald , “A Joint Model of Text and aspect ratings for Sentiment summarization” , Association for Computational Linguistics 2008.

[15] Arjun Mukherjee , Bing Liu,“Modeling Review Comments” , Association for Computational Linguistics, 2012