An Ontology based Anatomy Approach to Temporal Topic Summarization

A. Mekala,
Research Scholar, MCA, MSC, MPhil
Manonmaniam Sundaranar University, Thruneveli.

C.Chandra Sekar, PhD.
Reader, Department of computer science
Periyar University, Salem

ABSTRACT
Generally the searcher either searches for exact information based on the query or just surf topics which interest them on websites. Naturally, when user enters a query related to some topics, they did not get exact result of what they want. If the system selected the relevant passages, grouping together, made it summarizing and fluently, and returned the resulting text it will be an advantage to the user. Otherwise, if the resulting summary is not relevant enough to searcher, the user can refine the query. Thus, as a result, summarization is used as a technique for improving querying. To ensure this technique they proposed to summarize the content of a temporal topic in existing work by using an anatomy based summarization method called Topic Summarization and Content Anatomy (TSCAN). A temporal similarity (TS) function is implemented to generate the event dependencies and context similarity to form an evolution graph of the topic search. In this paper, we are combining two methods for topic summarization. The first method is mainly based on term-frequency, while the second method is based on ontology. We will construct an ontology database for analyzing the main topics of the article using NPL tool.

KEY WORDS: TSCAN, Ontology, Text summarization, Natural language processing

1. INTRODUCTION

Our approach follows what has been called a term-based strategy: find the most important information in the document(s) by identifying its main terms, and then extract from the document(s) the most important information (i.e., sentences) about these terms. In this project the number of documents will be given as input which will be the testing dataset. The training data set will be in database already. From these two datasets the conceptualization will be done from this we will get a set of summery result. Then the event segmentation and summarization method will be implemented. The NLP tool is used to eliminate the stop words and the stemming process will be done. After the required summarization result will be obtained the result will be displayed related to the searching article. The architecture diagram is given in Fig 1.

1.1 NLP TOOL OVERVIEW

Natural language processing (NLP) is also called as machine learning is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and natural languages. The natural language processing is a very attractive method of human–computer interaction. Modern NLP algorithms are introduced in machine learning, especially in statistical machine learning. Research into modern statistical NLP algorithms requires an understanding of a number of disparate fields, including linguistics, computer science, and statistics.

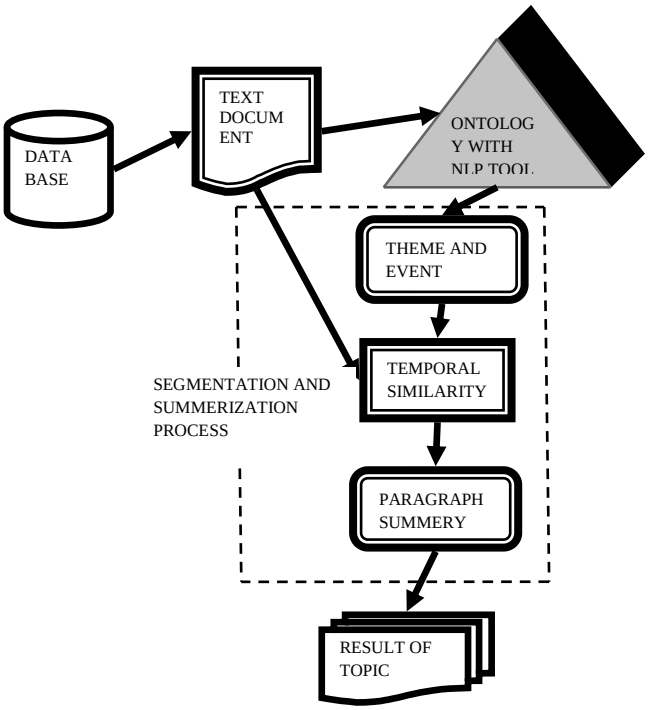

Fig 1: Architecture diagram

1.2 TOPIC ANATOMY

A topic is a real world incident that consists of one or more themes, which are related to a finer incident, a description, or a dialogue of a certain issue. Topic anatomy is an emerging text mining research issue that involves three major tasks: **Theme generation, Event segmentation and summarization, and Evolution graph construction.**

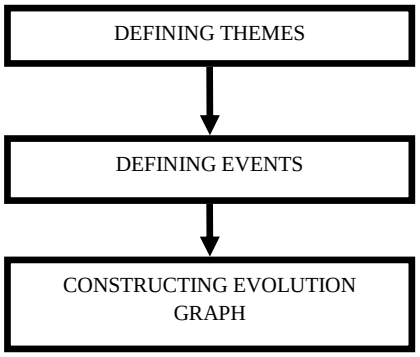

Fig 2: Flow view showing preferred embodiment

Defining themes: The content of a topic is comprised of several simultaneous themes, each representing an episode of the topic. The theme generation process tries to identify the themes of a topic from the related documents. A theme of a topic is derived from a collection of blocks.

Defining events: An event is defined as a disjoint sub-episode of a theme. The event segmentation and summarization process extracts topic events and their summaries by analyzing the intension variation of themes over time.

Constructing evolution graph: Context similarities of all of the events and themes are calculated and an evolution graph is formed by associating all of the events and themes according to the temporal closeness of each of the events and themes of the document. From this we can analyze the performance, precision, recall rate etc.., by comparing the existing system and proposed system.

1.3 TEXT SEGMENTATION

The main aim of text segmentation is to partition an input text into nonoverlapping segments such that each segment is a subject-coherent unit, and any two adjacent units represent different subjects. Depending on the type of input text, segmentation can be classified as story boundary detection or document subtopic identification. The input for story boundary detection is usually a text stream, e.g., automatic speech recognition transcripts from online newswires, which do not contain distinct boundaries between documents. Generally, naive approaches, such as using cue phrases, can identify the boundaries between documents efficiently. For document subtopic identification, the input is a single document, and the task involves identifying paragraphs in the document that relate to a certain subtopic.

Document subtopic identification enables many information systems to provide fine-grained services. Topic segmentation differs from document subtopic identification in a number of respects. First, the input for topic segmentation is a set of documents related to a topic, rather than a single document used in document subtopic identification. Second, the identified segments of a topic, i.e., the events of themes, have a temporal property rather than a textual paragraph or several contiguous paragraphs in a document. Finally, the segments of a document are disjoint textual units, but the events of a topic can overlap temporally.

1.4 TEXT SUMMARIZATION

Generic text summarization automatically creates a condensed version of one or more documents that captures the gist of the documents. As a document's content may contain many themes, generic summarization methods concentrate on extending the summary's diversity to provide wider coverage of the content. In this study, we focus on extraction-based generic text summarization, which composes summaries by extracting informative sentences from the original documents. Their proposed method [8] allows the user to search for specific types of information (for example, opinion, fact or encyclopedic knowledge).

Therefore, this proposed method produces summaries according to the type of information specified by the user as well as the topics of the documents. Text structure is also producing more balanced and coherent output summaries. The three main aspects of the problem in this dissertation are as follows: A. Extracting balanced contents of the source documents. B. Summarization to discriminate between types of information (fact, opinion, and knowledge) that the user's desire to know. C. Generating output summaries to improve the readability and reduce redundancy. We used text structure and document genre to extract the important sentences from the source documents in *A* and *B*, while we used text structure of output summaries to produce summaries in *C*.

1.5 ONTOLOGY

Applications of ontology-related techniques have become increasingly popular in recent years. Nevertheless, there is no unique definition of ontology in literature yet. We use Gruber's definition of ontology: "ontology is an explicit specification of some topics. It is a formal and declarative representation, which includes the vocabulary (or names) for referring to the terms in a specific subject area and the logical statements that describe what terms are, how they are related to each other." Essentially, the ontology decomposes the world into several objects for describing them. The determination of the way we describe objects and the formalism of representation depend on individual applications. In this paper, the ontology is designed for analyzing and gathering the semantic information of a class of article. Assuming every article contains several subtopics; we use the ontology for identifying subtopics of articles, and encode each of these possible subtopics by a non-overlapping portion of the ontology.

1.5.1 THE NEED FOR ONTOLOGY

We notice that all the above mentioned work assumes that all information provided by different sources to be integrated is covered by a domain model. However, information is not necessarily presented in the same way. Due to this fact, information exchange is not an easy task if different actors (producers or consumers of information) have not agreed on the semantic of data. It is necessary then to define an "alphabet" to ensure a good interpretation and understanding of exchanged data. The role of the ontology is to provide a common model that ensures the minimal requirements for this purpose. In fact, such a model allows one to construct a common view of different sources.

The elements in the model are described in a way independent from the particularity of the data source. One has to note that the more an application domain is restricted, the more it is possible to elaborate a precise description of the domain with the help of an ontology, and the more the processing may be refined. This is achieved mainly with the help of a domain's meta-data. Ontology is an explicit specification of some topic. Ontology is a way to discompose a world into objects, and a way to describe these objects. This is a partial description of the world, depending on the objectives of the designer and the requirements of the application or system. For each domain, there may be a number of ontologies. The use of ontology differs from an application to another, so are its design and its formalism of representation.

1.5.2 MODELING OF ONTOLOGY

Our intended use of ontology is to describe a data model, rather than knowledge. Therefore, it is not necessary now to include inference and reasoning mechanism to produce new knowledge. We will use a modeling of ontology close to object-oriented (OO) modeling. We believe that with OO paradigm, we can express ontology in an explicit way and generate software elements that are easily exploitable by other

applications. We propose a design of ontology that uses a 3-level model: basic objects, meta-model, model. The metamodel layer has been introduced and experimented where the goal is to describe axioms of ontology. We use the same concept here, but in a more general framework. We use this approach to express specific design needs of the user. Particularly, we will allow a redefinition of the roles of the elements of the model by the designer, according to the particular requirements of the application. Each user can specify his own metamodel of his ontology.

2. RELATED WORK

Naturally documents will be retrieved if and only if they contain keywords specified by the searcher in the websites. However, many documents contain the desired semantic information, even though they do not contain user specified keywords. This limitation can be addressed through the use of query expansion mechanisms. Additional search terms are added to the original query based on the statistical co-occurrence of terms [16]. In order to overcome the problem of keyword-based technique in responding to information selection requests we have designed and implemented a concept-based model using ontologies [13, 19, 20]. This model, which introduces a domain dependent ontology, is presented in this paper. Ontology is a collection of concepts and their interrelationships which can collectively provide an abstract view of topics related to the topic search [11, 12].

There are two different tasks for the ontology-based model: one is the extraction of semantic concepts from the keywords of what the user request to search the topics and the other is the actual construction of the ontology to improve the mining performance. With regard to the first problem, the key issue is to identify appropriate concepts that describe and identify documents. In this it is important to make sure those irrelevant concepts will not be associated and matched, and that relevant concepts will not be discarded. With regard to the second problem, we would like to construct ontologies automatically. In this paper we address these two problems together by proposing a new method for the automatic construction of ontology. From this the related topic search only will be mined from that the text summarization and content anatomy model will be implemented using NLP tool which produce the exact mining result and accurate results.

Historically ontology has been employed to achieve better precision and recall in the text retrieval system [13,15]. Here, attempts have taken two directions, query expansion through the use of semantically related-terms, and the use of conceptual distance measures [14, 15, 16, 17]. For the construction of ontology, the above papers assume manual construction; however, only a few automatic methods are proposed [17, 21, 22]. Elliman et al. [21] propose a method for constructing ontology to represent a set of web pages on a specified site. Self organizing map is used to construct hierarchy. Bodner et al. [17] propose a method to construct hierarchy based on statistical method (frequency of words). Hotho et al. [16] propose various clustering techniques to view text documents with the help of ontology.

Note that a set of hierarchies will be constructed for multiple views only; not for ontology construction purpose.Kleinberg [23] developed a topic evolution mining technique that constructs a hierarchical tree from a series of topic documents. The technique utilizes a HMM-based, two-state transition diagram to model the status of topics and splits a topic into diverse themes, modeled as tree branches, if the topic contains bursty information. Nallapati et al. [24] formalized the problem of topic evolution mining as a text clustering task in which the identified clusters, i.e., the events of a topic, are connected chronologically to form an evolution graph of the topic. In addition to constructing a graph, Mei and Zhai [1] modeled the activeness trend of identified themes.

As the trend reveals variations in the activeness of a theme over the lifespan of a topic, it helps users follow the evolution of the topic and its subsequent decline. Yang and Shi [25] focused on the temporal properties of a topic, and showed that fine-grained evolution graphs can be obtained by using the temporal information about topics. Feng and Allan [26] proposed an incident threading method that is similar to the proposed TSCAN system. The method first identifies incidents (i.e., events) from news documents; then, the semantic dependencies between the incidents are examined to produce an incident network. The authors also defined hand-crafted rules and an optimization procedure to assign types to network links. Experiments show that link type assignment is a challenging task, and better modeling of natural languages is required to improve the technique's accuracy.

3. PREVIOUS WORK

The idea of incident threading was motivated by existing research topics. Topic Detection and Tracking (TDT) monitors a news stream and places the information pieces into individual topics, while each topic includes all the news events closely related. It is ignored in TDT how a topic is established by the news events, and event threading tries to capture the internal structure of these topics. In addition to the effort of automatic news organization, discourse analysis studies the information ow in a press article with manual analysis. There are other news processing tasks, like novelty detection, news summarization and information filtering, which also aim at helping users in their news browsing. We now summarize these works and show why they have not provided an ideal framework for news organization.

TDT is a research program that focuses on event-based news organization. It splits the incoming news stream into a list of topics where each topic is a set of news stories that are strongly related by some seminal real-world event [2]. To simplify the problem, several assumptions were made in TDT: Topics do not overlap which means each news story belongs to at most one topic. From our observation, it is not always true since parts of the same story are often about different events, and sometimes the boundary between similar topics is not very clear. Topics are independent which means that a topic is a complete object and any relation to others is ignored, which is obviously not the case in reality. Topics are indivisible which denotes that all the evaluation metrics are topic-based, and participants of TDT place all the efforts into making their topics closer to the truth. However, an important factor is ignored in this assumption.

During the lifespan of a topic, one theme may attract more attention than the others, and is thus reported by more documents. We define an event as a significant theme development that continues for a period of time. Naturally, all the events taken together form the storyline of the topic. Although the events of a theme are temporally disjoint, they are considered semantically dependent in order to express the development of the theme. Moreover, events in different themes may be associated because of their temporal proximity

and context similarity. The proposed method identifies themes and events from the topic's documents, and connects associated events to form the topic's evolution graph. In addition, the identified events are summarized to help readers better comprehend the storyline of the topic.

4. AN ONTOLOGY BASED ANATOMY APPROACH

In this work first, we collect vocabularies and synonyms. Then, we put those terms by the Data model of ontology. The first step of ontology based TSCAN approach is to determine the by comparing the terms of documents with terms in the ontology. If the term does not exist in the ontology, we ignore it. Otherwise, we record the number of times the word appears in the ontology. The ontology decomposes the specific domain into several objects for describing them. The determination of the way we describe objects and the formalism of representation depend on individual applications. In this paper, the ontology is designed for analyzing and gathering the semantic information of a class of article.

Assuming every document contains several subtopics; we use the ontology for identifying subtopics of document, and encode each of these possible subtopics by a non-overlapping portion of the ontology. After selecting the blocks using ontology we construct symmetric block association matrix and finding the Eigen value from that matrix. Many different contents and structures exist in constructed ontologies, including those that exist in the same domain. If extant domain ontologies can be used, time and money can be saved. However, domain knowledge changes fast. In addition, the extant domain ontologies may require updates to solve domain problems. The reuse of extant ontologies is an important topic for their application. Thus, the integration of extant domain ontologies is of considerable importance. Through the method, two extant ontologies can be converted into a fuzzy ontology. The new fuzzy ontology is more flexible than a general ontology. The experimental results indicate that our method can merge domain ontologies effectively. A temporal similarity (TS) function is applied to generate the event dependencies and context similarity to form an evolution graph of the topic.

4.1 Construct ontology

In this module, first we will collect vocabularies and synonyms. Next, we put those words by the Data model of ontology. The first step of our method is to determine the main subtopics of the article of interest. This is achieved by comparing the words of articles with terms in the ontology. If the word does not exist in the ontology, we ignore it. Otherwise, we record the number of times the word appears in the ontology we encode the ontology with a tree structure, and each node includes the concepts represented by the node's children. When the count of any node increases, the counts associated with their ancestors will also in-crease.

After marking the counts of the nodes in the ontology, we select second-level nodes that have higher counts as the main subtopics of the article. Generally speaking, one article is composed of several subtopics, so our system will select multiple subtopics. There are limited topics an article can contain, and a reasonable summary probably should include fewer. Therefore, we only choose a limited number of subtopics and ignore others. We choose to ignore the subtopic if its count is less than 10. In addition, we choose only top three or required subtopics. After obtaining the subtopics, our system will use them for selecting paragraphs as the summary.

5. CONSTRUCTIONG ONTOLOGY

5.1 ONTOLOGY MAPPING ALGORITHM

The process uses the following functions or data structures to check whether two given concepts meet the predefined rules.

initialize: Initialize the process.
next-c: Request a particular OA for the next concept of a specified ontology. It returns the concept if it exists.
next-a: Request a particular OA for the attribute of a specified concept. It returns the attribute if it exists.
next-t: Request a particular OA for the corresponding data type of a certain attribute of a specified concept. It returns the data type if it exists.
comp: Compare two items. It returns true if two compared items are equal. Otherwise, it returns false.
syn: Request the SA for the synonyms of a given concept. It returns a synonym list (e.g.syn-list) if it exists.
measure-sim: Compute the weighted similarity over the given methods
add: Add items to the mapping results.
next-i: Request an equivalent relation from the mapping results.
add-sub-concept: Add inclusive relations to the mapping results.

Pseudo code of ontology mapping algorithm

Assume the mapping algorithm starts from a given start point.
O_1; O_2: source ontology and target ontology, respectively
c_1; c_2: concepts from these given ontologies, respectively
att_1; att_2: attributes of *c1; c2*, respectively
$type_1$; $type_2$: data types of att_1; att_2, respectively
$flag_1$; $flag_2$: flags with Boolean values
syn-list: a list of synonyms of a specified concept label returned from function *syn*
w_1; w_2; w_3: weights for specific methods
sim_1; sim_2; sim_3; sim: similarities
mapping-results: a data repository to record mapping results
δ: given threshold to filter out inappropriate mappings

```
Function mapping {
initialize;
do{
c1=next-(O1);
c1=next-c (O1);
if! (comp(c1.label,C2.label)){
syn (c1.lable);
}else{
sim1=measure-sim (R1);
}//end if
flag1=flag2=true;
while(syn-list(c1.label)!=Null),
{if!(comp(syn-list(c1.label),c1.label)){
flag1=false;
}else{
sim2=measure-sim(R2);
}//end if
if!(flag1){while(next-a(c1)!=Null&&next-a(c2)!=Null){
  att1=next-a(c1);
   att2=next-a(c2);
  type1=next-t(c1);
  type2=next-t(c1);
   if!(comp(att1,att2)&&comp(type1,type2)){
```

flag$_2$=false;
}else{
sim3=measure-sim(R_3);
}//endif
}//end if (flag$_1$)
if (flag$_2$){
if ((next-a(c1)!=Null)&&(next-a(c_2)==Null)){
add(mapping-results,(c_1,c_2⊒));
}//end if
}//end if (flag$_1$)
if (flag$_2$){
if((next-a(c_1)!=Null)&&(next-a(c_2)==Null)){
add(mapping-results,(c_1,c_2,⊑));
}//end if
}//end if (flag$_2$)
sim= $\sum_{j=1}^{3}$measure-sim(R_j);
if sim ≥ δ{
add(mapping-results,(c_1,c_2,=));
}//end if
}//end while
while (next-c(O_1)!=Null&& next-c(O_2)!=Null);//end do
while (next-i!=Null){
add-sub-concept;
}//end while
}//end function

5.2 ONTOLOGY INTEGRATION ALGORITHM

The algorithm uses the following functions or data structures to execute relevant operations.
initialize: Initialize the process.
next-c: Request a particular Ontology Agent for the next concept of a specified ontology. It
returns the concept if it exists.
search-r: Search for relations in mapping results. It returns existing relations if they exist or NIL otherwise.
copy-c: Copy a specific concept to the derived ontology properly.
insert-super: Insert a specified concept as a super-node of a given concept in a particular
ontology structure.
insert-sub: Insert a specified concept as a sub-node of a given concept in a particular ontology structure.
get-threshold: It returns the threshold.
check-consistency: Check the consistency of a specific ontology and return a Boolean value to indicate the current status of ontology consistency.
filter: Filter unexpected items from a given ontology, and return the filtered ontology.
The pseudocode of the ontology integration algorithm is shown below. This algorithm may execute repeatedly until the existing ontologies have been integrated as required.

Pseudocode of ontology integration algorithm
Assume the integration algorithm starts from a given start point.
O_1; O_2: two different ontologies to be integrated, respectively
O_d : the derived ontology
c_1, c_2: concepts from O_1 and O_2, respectively
l_{cd} : number of occurrences of concepts from O_d
m: a number of available ontologies relation: relations between given concepts which are from different ontologies
δ: threshold given by the user to filter inappropriate items from the generated ontology

Fuction integration{
initialize;
for(i=1;i<m;i++){
while((next-c(O_1)!=Null)&&(next-c(O_2)!=Null)){
c_1=next-c(O_1);
c_2=next-c(O_2);
relation=search-r(c_1,c_2);
switch(relation){
case"=";
l_{cd}++;
copy-c(c_d);
break;
case"⊑":
if(check-consistency){
insert-super(c_2,c_1);
l_{cd}=1;
}//endif;
break;
case"⊒":
if(check-consistency){
insert-sub(c_2,c_1);
l_{cd}=1;
}//end if
break:
default:
l_{cd}=1
}//end switch
}//end while
}//end for
threshold=get-thershold;
filter(O_d,δ);
}//end function

6. TOPIC MODEL

A topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. A topic is a real world incident that comprises one or more themes, which are related to a finer incident, a description, or a dialogue about a certain issue. During the lifespan of a topic, one theme may attract more attention than the others, and is thus reported by more documents. The proposed method identifies themes and events from the topic's documents, and connects associated events to form the topic's evolution graph. In addition, the identified events are summarized to help readers better comprehend the storyline(s) of the topic. A topic is represented explicitly by a collection of chronologically ordered documents. In this study, we assume that the documents are published in the same order as the events of the topic reported by independent authors, and that there is no inconsistency between the contents of the documents.

TSCAN decomposes each document into a sequence of non overlapping blocks. A block can be several consecutive sentences, or one or more paragraphs. We define a block as w consecutive sentences. For a topic, be a set of stemmed vocabulary without stop words. The topic can then be described by an m _ n term-block association matrix B in which the columns represent the blocks decomposed chronologically from the topic documents.

6.1 Theme Generation

A matrix, called a block association matrix, is symmetric matrix in which the entry is the inner product of columns i and j in matrix B. As a column of B is the term vector of a block, A represents the inter block association. Hence, entries with a

large value imply a high correlation between the corresponding pair of blocks. A theme of a topic is regarded as an aggregated semantic profile of a collection of blocks, and can be represented as a vector v of dimension n, where each entry denotes the degree of correlation of a block to the theme. Given the constitution of a vector computes the theme's association to the topic's content. The objective function of our theme generation process determines entry values so that the acquired theme is closely associated with the topic.

6.2 Event Segmentation and Summarization

The tasks of our event segmentation and speech endpoint detection are similar in that they both try to identify important segments of sequential data. In addition, it is the amplitude of sequential data that determines the data's importance. For example, given the speech utterance, the speech endpoint detection task involves distinguishing the significant segment S2 from the insignificant silent segments mixed with background noise. Here, S2 represents the word "one" and comprises a sequence of points with large positive and negative amplitudes. Therefore, we adopt Rabiner and Sambur's R-S endpoint detection algorithm for event segmentation. To segment events, the R-S algorithm examines the amplitude variation of an eigenvector to find the endpoints that partition the theme into a set of significant events. In the R-S algorithm, every block in an eigenvector has an energy value. To calculate the energy, we adopt the square sum scheme, which has proved effective in detecting endpoints in noisy speech environments.

STEP BY STEP ALGORITHM

Results ¬Æ; /*text summaries*/
Find all nodes *N= {N1,…,Nm}* that contain the keywords in *Q*; /**Ni* has the nodes that contain *wi**/
Repeat until the expanding areas of all combinations of nodes in *N1,…,Nm* meet.
{
For each node *v* in *N* do
{
Add to the expanding area of *v* the maximum-score adjacent edge from the (precomputed) shortest paths starting at *v* and ending at a node in *N* not containing the same keywords as *v;*

Check for new results (summaries) *T*; /*i.e., trees that contain a node from each of *N1,…,Nm* */
Trim summaries *T* to become minimal; calculate the score of *T*
Results;
Sort and output summaries in *Results*;
} }

6.3 Evolution Graph Construction

Automatic induction of event dependencies is often difficult due to the lack of sufficient domain knowledge and effective knowledge induction mechanisms. However, as event dependencies usually involve similar contextual information, such as the same locations and person names, they can be identified through word usage analysis. Our approach, which is based on this rationale, involves two procedures. First, we link events segmented from the same theme sequentially to reflect the theme's development. Then, we use a temporal similarity function to capture the dependencies of events in different themes. For two events, ei and ej, belonging to different themes, we calculate their temporal similarity between these two events and providing the graph description from the result.

7. EXPERIMENT RESULTS

Summary-to-document content similarity (SDCS) is defined as the average cosine similarity between an evaluated summary and the topic documents. Both components are represented by TF-IDF term vectors. A high similarity score implies that the summary is representative of the topic and can effectively replace the original topic documents for various information retrieval tasks. Parameter w controls the granularity of topic blocks. In the preprocessing phase of the experiments, we observed that the sentence segmentation program supplied by DUC sometimes segments sentences incorrectly when dealing with noun abbreviations followed by a period. The superior SDCS scores achieved by our method demonstrate the advantage of using event segmentation for temporal topic summarization.

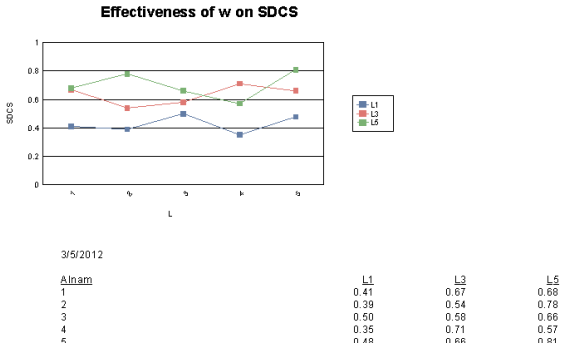

Fig 3: Effectiveness of w on SDCS

Although our summaries are not as diverse as those of the K-means method, they are more coherent. A popular measurement frequently used to judge the content coherence of a set of documents is the average pair wise document similarity TSCAN achieves superior APSBS (Average Pair wise Summary Block Similarity) scores. The reason is that our summaries focus on events in the first few significant themes; therefore, summary blocks have similar contexts. By contrast, the summaries compiled by other approaches try to cover diverse themes, so they are less coherent than our summaries. For all the summarization methods, APSBS decreases as the size of summaries increases. As a large summary covers many themes, its content is more diverse than that of a small summary. Hence, the average pair wise similarity will be low.

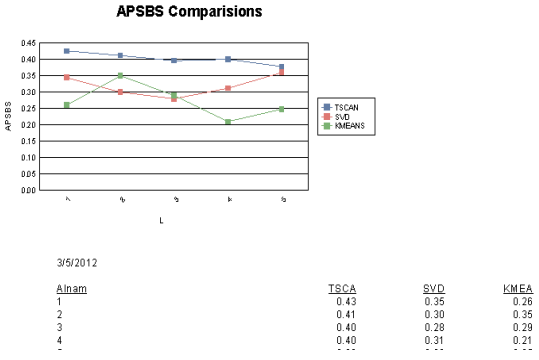

Fig 4: APSBS Comparison

8. RESULTS AND DISCUSSION

8.1 Precision Vs Recall rate

In this we are comparing the precision and recall rate for existing system and proposed system. When the precision rate is increased the recall rate is also increased in proposed system. The value for this graph is given below as Table 1:

SNO	Fraction of related document value	Ontology based TSCAN	TSCAN Approach
1	0.1	0.09	0.05
2	0.2	0.19	0.16
3	0.3	0.28	0.24
4	0.4	0.32	0.28
5	0.5	0.36	0.33

Table 1: Precision VS Recall

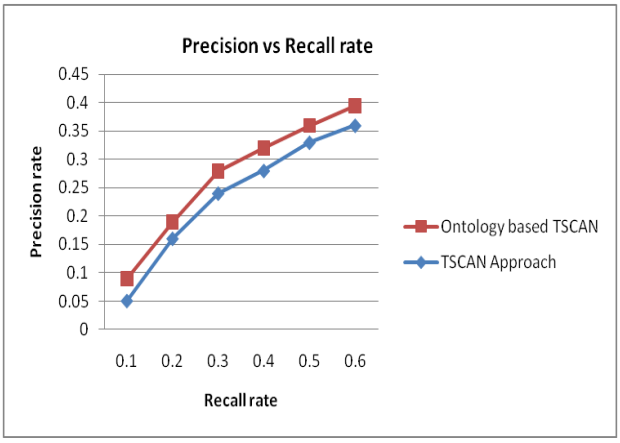

Fig 5: Precision Vs Recall rate

9. CONCLUSION & FUTURE WORK

The system we have built is a knowledge-based summarization system with the knowledge of topics coming from ontology. In this project, the ontology knowledge approach was presented, the approach based on feature appraisal and NLP application in summarization The knowledge is composed of not only in recognizing important topics in the document, but also in recognizing the relationships and the relationship types that exist between them. This extracted knowledge is represented in the form of evolution graph. Even without the summary, just looking at the nodes and relationships in the graph gives us an idea about what the document is taking about. A summary however gives us the actual details of the topic search.

This is the first system that uses ontological knowledge in this manner to obtain extractive summaries of topics. After identifying the main topics and determining their relative significance, we rank the paragraphs based on the relevance between main topics and each individual paragraph. Depending on the ranks, we choose desired proportion of Para-graphs as summary. Experimental results indicate that both methods offer similar accuracy in their selections of the paragraphs. In the future, we will research other method to determine the relationships between concepts more accurately instead of the above simple method and improve the method of ontology construction in a large data set.

REFERENCES

[1] Q. Mei and C.X. Zhai, "Discovering Evolutionary Theme Patterns from Text—An Exploration of Temporal Text Mining," Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining, pp. 198-207, 2005.

[2] J. Allan, editor. Topic Detection and Tracking: event-based information organization. Kluwer Academic Publishers, 2002.

[3] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Soren Auer, Christian Becker, Richard Cyganiak, Sebastian Hellmann. DBpedia - A Crystallization Point for the Web of Data

[4] Leonhard Hennig, Winfried Umbrath , Robert Wetzker. An Ontology-based Approach to Text Summarization, 2008.

[5] Xing Jiang, Ah-Hwee Tan. Learning and inferencing in user ontology for personalized semantic web search, Information Sciences, 2009, 2794-2808.

[6] Marek Obitko, Vaclav Snasel, Jan Smid. Ontology design with formal concept analysis, Edited by Vaclav snasel, Radim Belohlavek. In: Proc of the CLA 2004 Intl. workshop on Concept Lattices and their Applications Ostrava, Czech Republic, Sept. 2004, 111-119.

[7] Hele-Mai Haav. A semi-automatic method to ontology design by using FCA, Edited by Vaclav Snasel, Radim Belohlavek, In: Proc. of the CLA 2004 lnfl. Workshop on Concept Lattices and their Applications Ostrava, Czech Republic, Sept. 2004, 13-24.

[8] Lixin Han, Guihai Chen. A fuzzy clustering method of construction of ontology-based user profiles,Advances in Engineering Software, 2009, 535-540.

[9] Deryle Lonsdale, David W. Embley, Yihong Ding, Li Xu, Martin Hepp. Reusing ontologies and language components for ontology generation, Data & Knowledge Engineering, 2010, 318-330.

[10] Rung-Ching Chen, Cho-Tscan Bau, Chun-Ju Yeh. Merging domain ontologies based on the Word-Net system and Fuzzy Formal Concept Analysis techniques, Applied Soft Computing, 2011, 1908-1923.

[11] T. R. Gruber, "Toward Principles for the design of Ontologies used for Knowledge Sharing", in Proc. of *International Workshop on Formal Ontology*, March 1993.

[12] Y. Labrou and T. Finin, "Yahoo! as Ontology: Using Yahoo! Categories to Describe Documents," in Proc. of *The Eighth International Conference on Information Knowledge Management*, pp. 180-187, Nov 1999, Kansas City, MO.

[13] N. Guarino, C. Masolo, and G. Vetere, "OntoSeek: Content-based Access to the Web," *IEEE Intelligent Systems*, Volume 14, no. 3, pp. 70-80, 1999.

[14] Latifur Khan "Ontology-based Information Selection, "Ph.D. Thesis, University of South California, 2000.

[15] Nicola Guarino, Claudio Masolo, Guido Vetere. "OntoSeek: Content-Based Access to the Web". IEEE Intelligent Systems 14(3): 70-80, 1999

[16] A. F. Smeaton and V. Rijsbergen, "The Retrieval Effects of Query Expansion on a Feedback Document Retrieval System". *The Computer Journa*l, vol. 26, No.3, pp239-246, 1993.

[17] R. Bodner and F. Song, "Knowledge-based Approaches to Query Expansion in Information Retrieval," in *Proc. of Advances in Artificial Intelligenc*e, pp. 146-158, New York, Springer.

[18] W. Woods, "Conceptual Indexing: A Better Way to Organize Knowledge," *Technical Report of Sun Microsystem*s, 1999.

[19] L. Khan and D. McLeod, "Audio Structuring and Personalized Retrieval Using Ontology," in *Proc. of IEEE Advances in Digital Libraries, Library of Congres*s, pp. 116-126, Bethesda, MD, May 2000.

[20] L. Khan and D. McLeod, "Disambiguation of Annotated Text of Audio Using Ontology," in *Proc. of ACM SIGKDD Workshop on Text Minin*g, Boston, MA, August 2000.

[21] Dave Elliman, J. Rafael G. Pulido. "Automatic Derivation of On-line Document Ontology". MERIT 2001, 15th European Conference on Object Oriented Programming, Budapest, Hungary, Jun 2001.

[22] A. Hotho, A. Mädche, A., S. Staab, "Ontology-based Text Clustering," Workshop Text Learning: Beyond Supervision, 2001

[23] J. Kleinberg, "Bursty and Hierarchical Structure in Streams," Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 91-101, 2002.

[24] R. Nallapati, A. Feng, F. Peng, and J. Allan, "Event Threading within News Topics," Proc. 13th ACM Int'l Conf. Information and Knowledge Management, pp. 446-453, 2004.

[25] C.C. Yang and X. Shi, "Discovering Event Evolution Graphs from Newswires," Proc. 15th Int'l Conf. World Wide Web, pp. 945-946, 2006.

[26] A. Feng and J. Allan, "Finding and Linking Incidents in News," Proc. 16th ACM Conf. Information and Knowledge Management, pp. 821-830, 2007.