Terminology Evolution in Web And Text Mining Using Association Rules

Master's Project submitted to
The Department of Computer Science at Montclair State University

In Partial fulfillment of the requirements for a degree in Master of Science in Computer Science

Debjani Roychoudhury
Faculty Advisor: Prof. Aparna Varde
7th May, 2009

This research is in collaboration with the Max Planck Institute For Informatics, Germany

TABLE OF CONTENT

ABSTRACT

We work on the problem of time-aware query translation over text archives on the web. We have a stream of time-stamped documents in the form of newswire articles, blog posts, web-pages etc. In answering user queries over online text archives spanning long time periods, it is desirable that historical information be incorporated. For example, a query on Mumbai should automatically retrieve documents with its former name Bombay. Hence, temporal terminology evolution needs to be taken into account for providing better responses to queries.

In order to achieve this, we need to discover the concepts in text archives whose names change over time. This is precisely the goal addressed in the research. It focuses on the sub-problem related to the discovery of such concepts, which we call SITACs, i.e., Semantically Altering Temporally Identical Concepts.

We propose an approach based on association rules to solve this sub-problem. In data mining, Association Rule Mining is a popular method for discovering interesting relations between variables in a database. The analysis is based upon strong rules discovered in databases using different measures of interestingness [5]. Using these rules Agrawal et al.[1] introduced Apriori algorithm for discovering regularities between products in a transaction Association rule mining using the classical Apriori algorithm helps to discover the temporal evolution in terminology, serving as the basis for time-aware query translation. Issues addressed in this work include defining transactions with respect to text corpora to perform the mining, proposing suitable measures for ranking the rules obtained and addressing the properties of rules such as transitivity. The details of the solution approach along with evaluation over real text corpora spanning historical archives are described in the paper. This evaluation presents several interesting rules capturing the temporal terminology changes.

INTRODUCTION

1.1 Background and Motivation

Time-stamped documents such as newswire articles, blog posts and other web-pages are often archived online. It is observed that when these archives cover long spans of time, the terminology within them could undergo significant changes. Hence, when users pose queries over such documents, the queries need to be translated taking into account these temporal changes, in order to provide more accurate responses to users. In other words, the query translation over the documents spanning such historical text archives needs to be time-aware.

We present a few motivating examples of queries over historical archives of text.

1. When was the USA formed by the Articles of Association?
2. When did the USA become independent through the Declaration of Independence?
3. How many states did the USA have at the time of independence?
4. When was the Constitution of the USA established?
5. How have the relations between the USA and the British Isles changed over the years?
6. What is the USA policy on Native Americans?

These queries would be entered on a search engine using appropriate keywords or sentences. In response to these queries, multiple text documents need to be referenced. An example of a document with answers to queries 1 and 2 is Lincoln's presidential address [16]. However, he refers to the USA as the Union. Query 3 can be answered from speeches of many presidents [16] who use terms such as British Isles and Great Britain in referring to the UK.

Note that this is not just an issue of synonymy, e.g., the terms USA and Union would not be detected in the literature as obvious synonyms. However, from a study of history, it is known that when the former presidents referred to the Union, they meant the United

States of America or the USA, as of today. We refer to such terms as SITACs, defined below.

1.2 Problem Definition

In order to describe the problem addressed in this paper, we first explain the term SITAC.

" Definition of SITAC: The term SITAC is an acronym for a Semantically Identical Temporally Altering Concept. It refers to a concept whose names change over time, although they in principle refer to the same entity. SITACs could be under different categories such as person names, places, organizations, item names and more. "

Examples of SITACs include:

- *Person:*
  - Margaret Thatcher, Margaret Roberts
  - Mother Teresa, Agnes Gonxha Bojaxhiu
- *Place:*
  - Kalikata; Calcutta; Kolkata
  - St. Petersburg, Leningrad, Petrograd
  - Mumbai, Bombay
- *Organization:*
  - Virginia Agricultural and Mechanical College and Polytechnic Institute; Virginia Polytechnic Institute; Virginia Polytechnic Institute and State University (Virginia Tech)
  - AT&T; Lucent Technologies; Alcatel-Lucent (the parent organizations of Bell Labs)
- *Item:*
  - iPod; MP3 Player; Walkman

Given the definition and examples of SITACs, we now outline the aim of our main problem relating to terminology evolution.. In our overall problem, we address the following three goals:

1. Given text corpora, discover the SITACs, i.e., Semantically Identical Temporally Altering Concepts.
2. Answer queries accordingly by using the SITACs, thus making the query translation time-aware.
3. Rank responses using domain semantics and temporal factors.

In this paper, we focus on the first sub-problem namely the discovery of the SITACs. More specifically, the goal of this paper is as follows:

Goal of SITAC Discovery Sub-problem: To identify the SITACs over text corpora and store them in databases for future use, setting the stage for time-aware query translation.

We present a solution to this sub-problem of SITAC discovery, addressing several issues involved and implementing our solution over real historical archives.

PROPOSED SOLUTION

Our proposed approach to solve the sub problem of SITAC discovery is based on association rule mining. Association rule mining looks for interesting relationships among items in a given data set [9]. This concept can easily be described using "Market Basket Analysis". If a customer purchases computer also tend to purchase software at the same time. So in a computer store, placing hardware and software can influence the buying strategy of a customer. So we are interested to get the items that are frequently associated or purchased together. These patterns can be represented by association rule. The interestingness of a rule can be measured by rule support and rule confidence. A rule can be considered as interesting if it satisfies the minimum threshold and minimum confidence threshold. Aprioi is a strong algorithm for mining frequent item sets. This algorithm is based on the fact that algorithm uses a "prior knowledge" of associated items[1].

The approach has three steps, namely, Document and Concept Extraction, Rule Derivation and Ranking of Rules.. These are described below.

2.1 Document and Concept Extraction

The archived text sources given to us need to be preprocessed such that we get information in the form of documents and concepts (terms) over time. Thus, we have the following.

- *Document:* Text source, denoted as d with time-stamp t
- *Concept:* Individual term (word or phrase), denoted as c

We thus extract concepts from time-stamped documents in the text archives. The concepts are primarily nouns and noun phrases as they refer to various entities such as person, place, organization and item.

Note that there is some natural language processing involved here, mainly in the form of noun phrase chunking. This will be elaborated in the section on experimental evaluation which includes a description of our implementation.

2.2 Rule Derivation

We need to discover rules of the type *(c1, t1) => (c2, t2)* that capture temporal relationships over concepts. These rules can be derived from the corresponding time-stamped documents. We use the Apriori Algorithm to mine the association rules, for which we need to define transactions with respect to the text archives. We first consider a Document Transaction.

Definition of Document Transaction: A Document Transaction in our context is denoted as X and is defined as a set of correlated documents. Documents are said to be correlated if they are referenced in a single query.

Consider *n* to be the total number of Document Transactions and *m* to be the total number of documents. We thus consider each document transaction *Xi* as follows, where *Y/N* denotes presence or absence of documents in the Document Transaction.

X1 : [d1, Y/N], [d2, Y/N] . . . [dm, Y/N]

X2 : [d1, Y/N], [d2, Y/N] . . . [dm, Y/N]

. . .Xn : [d1, Y/N], [d2, Y/N] . . . [dm, Y/N]

Using this information on Document Transactions, we then need to store the data in the form of concepts within each document. Consider *m* to be the total number of documents and *p* to be the total number of concepts. We now define a Concept Transaction as follows.

Definition of Concept Transaction: A Concept Transaction Di in our context is defined as a collection of concepts within a given document di..

Thus, we have the following concept transactions, where *Y/N* denotes the presence or absence of concepts within the respective documents and hence within the Concept Transaction.

D1 : [c1,Y/N],,[c2, Y/N] ....[cp,Y/N]

D2 : [c1,Y/N], [c2, Y/N] ....[cp,Y/N]

... Dm : [c1,Y/N], [c2, Y/N] ....[cp,Y/N]

From this information on Concept Transactions and Document Transactions, we construct Document-Concept Transactions defined as follows.

Definition of Document-Concept Transaction: A Document- Concept transaction Ti is a set of correlated documents along with the concepts contained within them.

Consider a simple example where a Document Transaction *X1* consists of documents *d1* and *d2*. Thus, we have:

X1 : [d1, Y], [d2, Y], [d3,N] . . . [dm,N]

Considering only the documents that are present and omitting the *(Y/N)* notation, this translates to:

X1 : [d1], [d2]

Assume for simplicity that document d1 has concepts c1 and c4, while document d2 has concepts c2, c3 and c5. Thus, the corresponding Document-Concept Transaction would be:

T1 : [D1: c1,c4], [D2: c2,c3,c5]

Where concepts *c1* and *c4* come from document *d1* (i.e., Concept Transaction *D1*) and thus bear its time-stamp, *t1,* while concepts *c2, c3* and *c5* come from document *d2,* (i.e., Concept Transaction *D2*) and hence bear its time-stamp, *t2.*

Likewise, we store the Document-Concept Transactions. Thereafter, using suitable minimum support and minimum confidence thresholds, the Apriori algorithm can be applied to derive association rules. Since each document has a time-stamp, a rule such as *(c1,d1) => (c2,d2)* obtained from the data would automatically mean *(c1,t1) => (c2,t2),* thereby depicting temporal changes, and identifying SITACs *c1* and *c2.*

2.3 Ranking of Rules

The rule derivation step can lead to several association rules being mined. In order to emphasize the really interesting SITACs with reference to context, the derived association rules need to be ranked in the order of importance. For this, we define a measure called correlation score as the strength of the relationship between any two concepts with reference to context. It incorporates fundamental similarity among concepts and their frequency of co-occurrence. This score is denoted as *S(c1,c2)* and calculated as follows.

The fundamental similarity between concepts *c1* and *c2* refers to their closeness in terms of dictionary meaning and domain semantics. We measure this in terms of the similarity between the concepts employing the widely used Jaccard's Coefficient [10]. We refer to this as ontological similarity *O(c1,c2).* We deploy the notion that the closer the concepts are from each other with respect to ontological similarity, the more similar they are with reference to context, giving a higher correlation score.

Thus, $S(c1,c2) \propto O(c1,c2)$.

Next, we consider the frequency of co-occurrence of the concepts. Our argument is that if two concepts c1 and c2 often co-occur in queries, they are more likely to be related than other concepts that are never seen together. We refer to this type of co-occurrence as concept intersection I(C1,C2). It is measured as the frequency with which concepts occur together in a single query, considering anticipated user queries. Based on this argument, the greater the concept intersection, i.e., frequency of co-occurrence, the higher is the correlation score.

Thus, $S(c1,c2) \propto I(c1,c2)$.

Combining the two, we get:

$S(c1,c2) \propto I(c1,c2).O(c1,c2)$

Thus, $S(c1,c2) = k \times I(c1,c2) / O(c1,c2)$ where k is a proportionality constant.
This correlation score $S(c1,c2)$ is used to rank the derived temporal association rules. Domain knowledge can be further applied to prune obvious and uninteresting rules.

Thus, the three steps of Document and Concept Extraction, Rule Derivation and Ranking of Rules describe our proposed approach for SITAC discovery. We now present its evaluation.

EXPERIMENTAL EVALUATION

We discuss the implementation of our approach along with a summary of our experimental evaluation. Details are available in our technical report [8].

3.1 Implementation

The overview of implementing our approach is illustrated in Figure 1. Using given text corpora, we performed natural language processing with state-of-the-art tools to preprocess the text archives.

Among available freeware tools, we used the Stanford Parser for sentence parsing and the Minipar NP chunker for Noun Phrase extraction. We further wrote shell scripts to extract the documents and concepts from the text archives.

These were used as the input to the widely-used data mining tool WEKA, i.e., the Waikato Environment for Knowledge Analysis, which includes the code for implementing the Apriori algorithm for association rules.

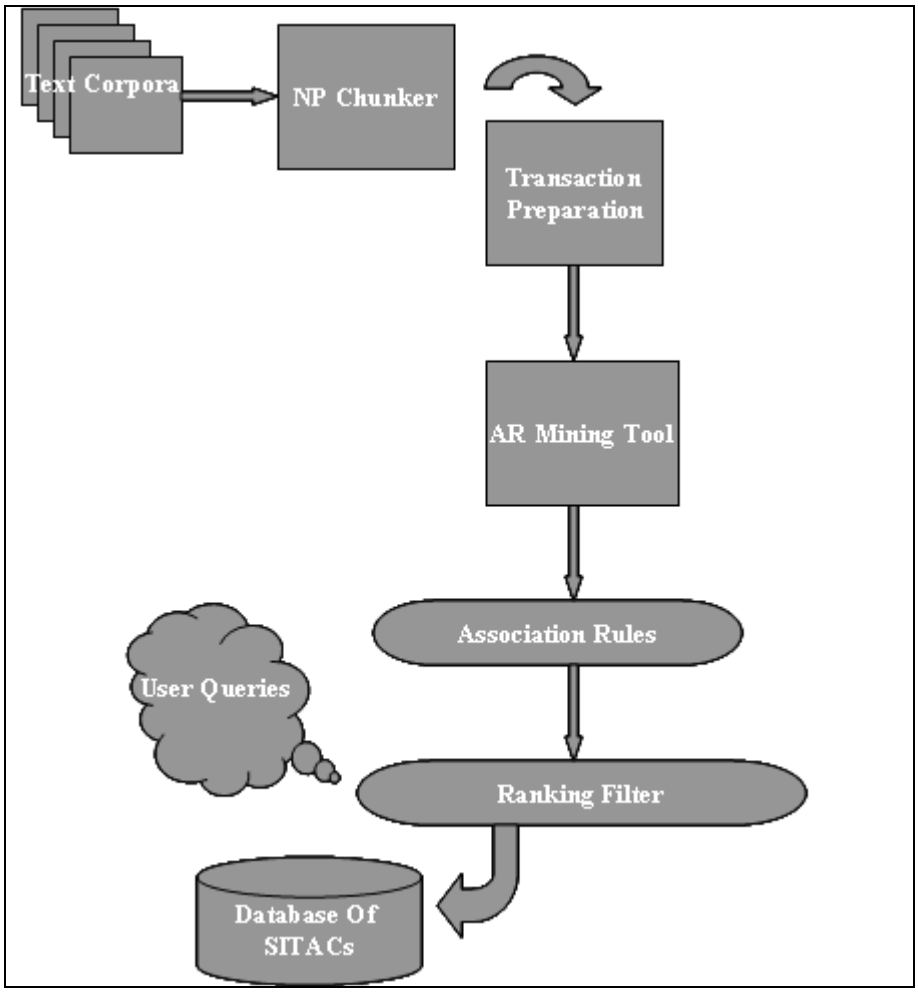


Figure 1: Overview of our Approach

3.2 Preprocessing of Data

Data Preprocessing was an important step in our experiments. We experimented with a text corpus containing Lectures of American Presidents that spanned across the 200 years of American History. The purpose was to discover significant relationships among concepts and keywords embedded in the lectures. Our technique for extracting Rules was essentially based on WEKA software. The method was general, but required appropriate format for each file which can be readable by WEKA. The method for generating Rules has three phases: extracting keywords, checking the frequencies of all keywords and save all the results in a special format file which will be readable by WEKA. WEKA can take

CSV files and ARFF files. So the collected data must be preprocessed so as to make it compatible with WEKA formats.
WEKA has an in-built converter program which converts text files to ARFF files. But we did not utilize the converter software as we found that it requires some manual pre processing of the data.

Challenges that we have faced with Pre Processing:

The main challenge was to formulate the special format files with desired keywords and frequency count. But CSV file can only be processed if we provide the range instead of numeric figures. We tried to formulate the ARFF file with frequency count but WEKA could not open the file containing numeric values. We tried to use range of values instead of a particular frequency. WEKA could read range of values but the Rules came out very meaningless.

3.3 Experimentation

After preprocessing data we have started our experiments. The purpose was to collect rules from the Gutenberg Text Corpus. We have followed three different approaches to gather the rules.

A. Manual Approach

Method:

Manual Approach involved following steps:

- We have divided the text corpus into different text files based on the President's Lectures.
- We created total five text files for different periods of time (1790 – 2007).

- Then we manually searched the Noun keywords to get relevant rule set. To perform this step we have used the sample queries from Time Aware Query Expansion over Text Archives by *Aparna Varde , Gerhard Weikum, Klaus Berberich, and Srikanta Bedathur*
- In our experiments, we eliminated the stop words. Our aim was to explore the relations between Noun keywords. So to avoid noise, we eliminated stop words from our search. Some examples of stop words are as below:

  Anybody, Become, Another, And, Both, Even, Full etc.

- All the keywords were saved in special text files (.arff format).
- WEKA can take .arff file as an input file and can produce the rule set after analyzing the attributes and values.

Rule Result:

Best rules found:

1. Constitution=Y 4 ==> Union=Y 4 conf:(1)
2. Union=Y 4 ==> Constitution=Y 4 conf:(1)
3. United-States=Y 4 ==> Union=Y 4 conf:(1)
4. Union=Y 4 ==> United-States=Y 4 conf:(1)
5. Government=Y 4 ==> Union=Y 4 conf:(1)
6. Union=Y 4 ==> Government=Y 4 conf:(1)
7. United-States=Y 4 ==> Constitution=Y 4 conf:(1)
8. Constitution=Y 4 ==> United-States=Y 4 conf:(1)
9. Government=Y 4 ==> Constitution=Y 4 conf:(1)
10. Constitution=Y 4 ==> Government=Y 4 conf:(1)

Limitations:

We have got very interesting rules by using Manual Approach but this method was very time consuming and laborious. The whole experiment was done on a very small amount of data. We were able to take only five Lectures from Gutenberg text corpus. But our goal was to review as many lectures as possible to get the good amount of Rule sets. Otherwise experiments would not be authentic and informative.

B. Stanford Parser Approach

Stanford Parser is a natural language parser; it is a program that analyzes the grammatical structure of sentences. This is a statistical parser that uses the knowledge of probability. Probabilistic parsers use knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of every sentences.
Stanford Natural Language Processing Group has invented the parser in 1990.

This package is a Java implementation of probabilistic natural language parsers, both highly optimized PCFG and lexicalized dependency parsers, and a lexicalized PCFG parser. It supports a wide range of languages, English, German, Chinese, Arabic etc.

I have downloaded the software from their website -
http://nlp.stanford.edu/software/lex-parser.shtml

After downloading and installing the parser, I have used my text corpus as input files. It has a very nice GUI interface.

Method

Stanford Parser Approach involved following steps:

- The input files were the same text corpus that have been created at the time of manual approach.
- We have chosen EnglishPCFG parser to parse the sentences.
- The next step was to load input file using the GUI interface of the parser.
- After loading the entire file, we have randomly selected some sentences and parsed them using EnglishPCFG parser.
- The parser gives the output in a tree format with the list of all Noun/Adjective/Verb/Adverb phrases in the statement.
- From all the keyword listed by the parser, we had to choose Noun keywords that may form some interesting rules.
- After choosing the Nouns, we created arff files so that WEKA can analyze the file and we can apply Apriori Association rules in that arff files.

  The advantage of using the Stanford parser was that the searching of Noun keywords was much easier. We were able to review increased number of lectures.

  Below are the Stanford parser screen output and some rules that WEKA has found using Association Rule.

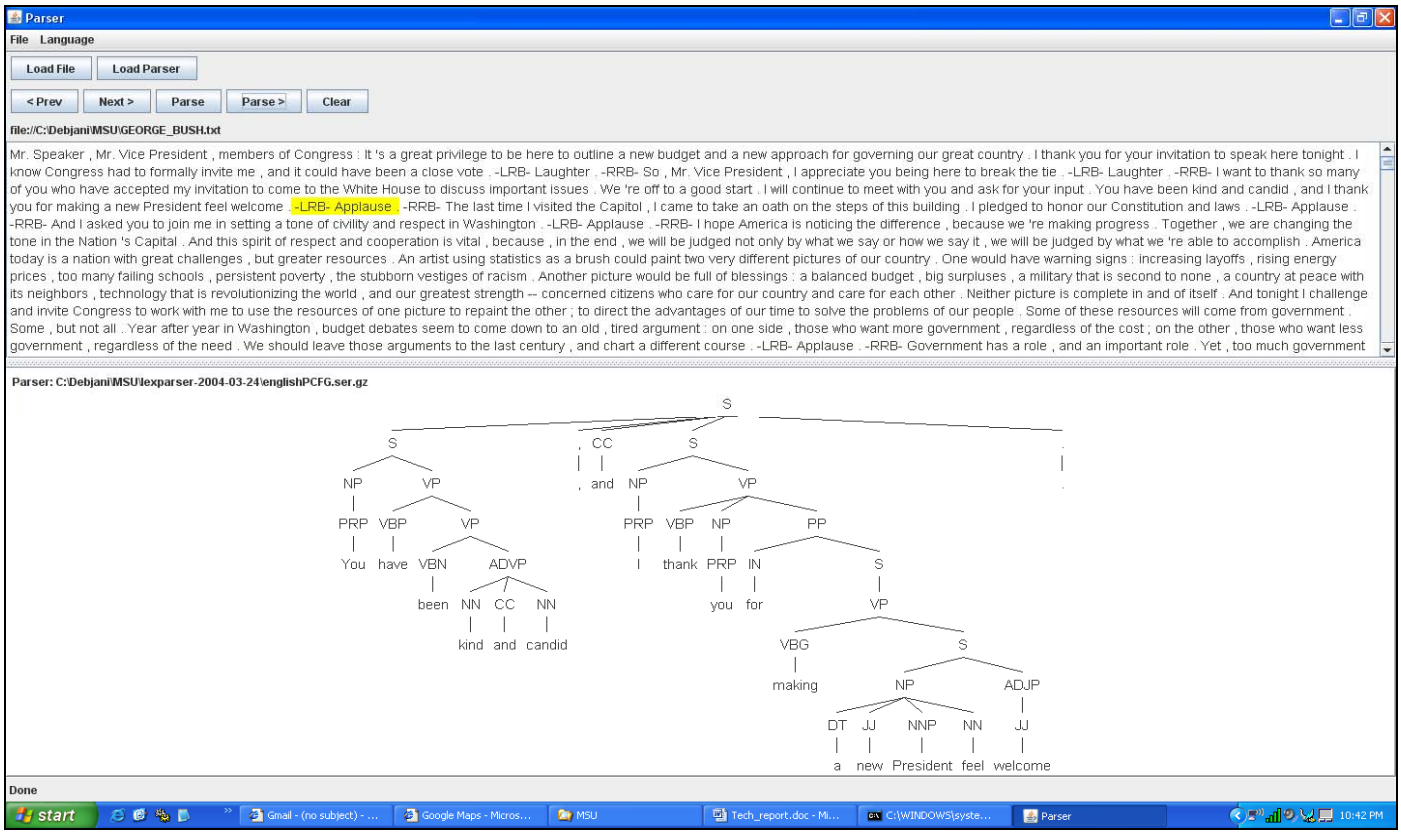


Figure2: Whole hierarchy of the grammatical relations

Rule Results:

1. Union=Y 5 ==> World=Y America=Y Government=Y Investmen=Y Freedom=Y Congress=Y 5 conf:(1)
2. World=Y 5 ==> Union=Y America=Y Government=Y Investmen=Y Freedom=Y Congress=Y 5 conf:(1)
3. America=Y 5 ==> Union=Y World=Y Government=Y Investmen=Y Freedom=Y Congress=Y 5 conf:(1)
4. Government=Y 5 ==> Union=Y World=Y America=Y Investmen=Y Freedom=Y Congress=Y 5 conf:(1)
5. Investmen=Y 5 ==> Union=Y World=Y America=Y Government=Y Freedom=Y Congress=Y 5 conf:(1)
6. Freedom=Y 5 ==> Union=Y World=Y America=Y Government=Y Investmen=Y Congress=Y 5 conf:(1)
7. Congress=Y 5 ==> Union=Y World=Y America=Y Government=Y Investmen=Y Freedom=Y 5 conf:(1)

8. Union=Y World=Y 5 ==> America=Y Government=Y Investmen=Y Freedom=Y Congress=Y 5 conf:(1)

9. Union=Y America=Y 5 ==> World=Y Government=Y Investmen=Y Freedom=Y Congress=Y 5 conf:(1)

10. Union=Y Government=Y 5 ==> World=Y America=Y Investmen=Y Freedom=Y Congress=Y 5 conf:(1)

Limitations

The biggest limitation that we came across with Stanford Parser is that it is very slow with big sentences. Sometimes it got hung if the sentences are 4 to 5 lines long.

Another problem was that it does not allow us to save the output in a text file. It is easier to extract only Noun Phrases from a text file. Since it is important to save the Stanford parser output in image format, this program could not be used to search for Noun words and manual searching had to be done..

C. Minipar NP Chunker Approach

MINIPAR is a broad-coverage parser for the English language. This system is for extracting typed dependencies of English sentences. In order to capture inherent relations occurring in texts corpus many NP relations are included in the set of grammatical relations used. The typed dependency extraction facility described here is integrated in the Minipar.

MINIPAR achieves about 88% precision with respect to dependency relationships. Minipar presents the grammatical relations as an output. For example, the dependent relation can be specialized to aux (auxiliary), arg (argument), or mod (modifier). The arg relation is further divided into the subj (subject) relation and the comp (complement) relation, and so on.

Other grammatical relations for NPs (amod – adjective modifier, rcmod - relative clause modifier, det - determiner, partmod - participial modifier, infmod - infinitival modifier, prep - prepositional modifier), our hierarchy includes the following grammatical relations: appos (appositive modifier), nn (noun compound), num (numeric modifier), number (element of compound number) and abbrev (abbreviation).

Method

We used MINIPAR NP chunker in a Linux Server Lemur.montclair.edu. Minipar has a very powerfull tool "pdemo. This program is a demonstration program. This program reads each line in the the standard input as a sentence and prints out the parse tree of the sentence. "pdemo" executable can be invoked with different options to do different types of analysis. For example:

```
-c
      Return the constituency tree instead of the dependency tree
-l
      Print the name of the grammatical relation between a node
and its
      Parent

-t
      Print the dependency triples instead of parse trees. Each
dependency triple consists of a head, a relationship and a
modifier, separated by a tab
```

After studying carefully all the options I tried to analyze the text corpus. This time the American President's Lectures were parsed in different text files.

The text files were analyzed by using the following command.

```
pdemo -c < INPUT_FILE >OUTPUT_FILE
pdemo -c </home/roychoudhud1/lincoln.txt>/home/roychoudhud1/lincoln.out
```

The output file thus had the Parse tree with grammatical relations among all the keywords. To extract the Noun Phrases, a small shell script was created which extracted only noun keywords from the output file produce another output file.

From this output file I have extracted similar keywords were extracted manually. The goal is to get some interesting rules.

The above process was repeated the above process for all the Inout text file which I created from American Presidents Lectures.

All the extracted keywords/concepts are saved in a excel document with CSV format. Weka can read the CSV files. All the CSV files were Time Aware so that we can easily associate rules with the time of the Presidents Lectures. The analysis of concepts are done by Presence and Absence for all the concepts in all the CSV files.

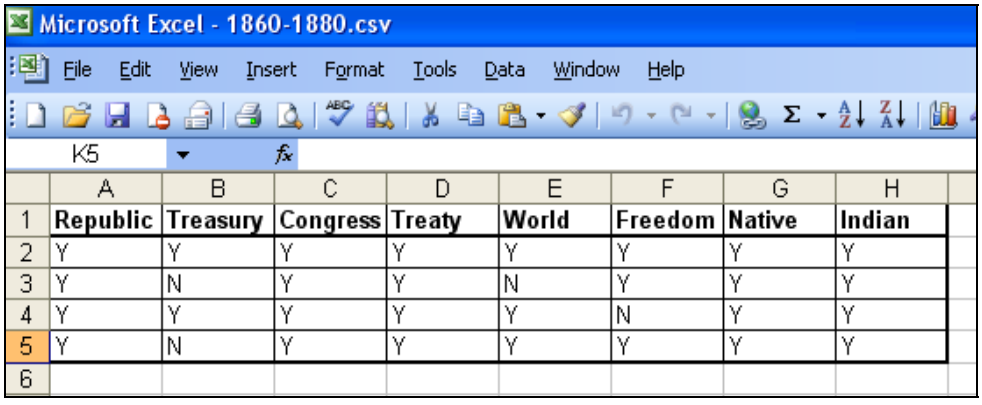
```
> (
1       (State  ~ N     *       )
2       (of     ~ Prep  1       mod     (gov state))
3       (the    ~ Det   5       det     (gov address))
4       (Union  ~ N     5       nn      (gov address))
5       (Address        ~ N     2       pcomp-n (gov of))
)
> (
1       (Abraham        ~ U     2       lex-mod (gov Abraham Lincoln))
2       (Lincoln        Abraham Lincoln N       *       )
)
> (
1       (December       ~ U     4       lex-mod (gov December 1, 1862))
2       (1      ~ U     4       lex-mod (gov December 1, 1862))
3       (,      ~ U     4       lex-mod (gov December 1, 1862))
4       (1862   December 1, 1862 N      *       )
)
> (
)
> (
1       (Fellow ~ U     3       lex-mod (gov Fellow-Citizens))
2       (-      ~ U     3       lex-mod (gov Fellow-Citizens))
3       (Citizens       Fellow-Citizens N       *       )
4       (of     ~ Prep  3       mod     (gov Fellow-Citizens))
5       (the    ~ Det   6       det     (gov senate))
6       (Senate ~ N     4       pcomp-n (gov of))
```

```
7       (and    ~ U     6       punc    (gov senate))
8       (House  ~ U     10      lex-mod (gov House of Representatives))
9       (of     ~ U     10      lex-mod (gov House of Representatives))
10      (Representatives        House of Representatives N      6
conj    (gov senate))
11      (:      ~ U     *       punc)

3       (last   ~ PostDet       4       post    (gov annual))
4       (annual ~ N     1       pcomp-n (gov since))
E0      (()     vpsc C  4       rel     (gov annual))
E1      (()     ~ N     E0      s       (gov vpsc)      (antecedent 4))
5       (assembling     assemble V      E0      i       (gov vpsc))
E4      (()     annual N        5       subj    (gov assemble)
(antecedent E1))
```

Figure 3: Minipar Output – Partial Screen Dump

All the extracted nouns (depicting concepts) were then converted into the required format for transactions as shown in the partial screen dump Figure 4. Each row in Figure 4 corresponds to a Document-Concept Transaction with presence or absence if concepts within the corresponding documents. Since the documents themselves are time-stamped, this implies that the corresponding rules derived from these transactions will incorporate the temporal aspect



	A	B	C	D	E	F	G	H
1	**Republic**	**Treasury**	**Congress**	**Treaty**	**World**	**Freedom**	**Native**	**Indian**
2	Y	Y	Y	Y	Y	Y	Y	Y
3	Y	N	Y	Y	N	Y	Y	Y
4	Y	Y	Y	Y	Y	N	Y	Y
5	Y	N	Y	Y	Y	Y	Y	Y
6								

Figure 3: Transaction Preparation – Partial Screen Dump

Rule Set 1: 1790-1810

1. United States=Y 4 ==> Civil War=Y 4 conf:(1)
2. Civil War=Y 4 ==> United States=Y 4 conf:(1)

3. Security=Y 3 ==> United States=Y Civil War=Y Patriotism=Y Indian-Tribes=Y 3 conf:(1)
4. Patriotism=Y 3 ==> United States=Y Civil War=Y Security=Y Indian-Tribes=Y 3 conf:(1)
5. Indian-Tribes=Y 3 ==> United States=Y Civil War=Y Security=Y Patriotism=Y 3 conf:(1)
6. United States=Y Security=Y 3 ==> Civil War=Y Patriotism=Y Indian-Tribes=Y 3 conf:(1)
7. United States=Y Patriotism=Y 3 ==> Civil War=Y Security=Y Indian-Tribes=Y 3 conf:(1)
8. United States=Y Indian-Tribes=Y 3 ==> Civil War=Y Security=Y Patriotism=Y 3 conf:(1)
9. Civil War=Y Security=Y 3 ==> United States=Y Patriotism=Y Indian-Tribes=Y 3 conf:(1)
10. Civil War=Y Patriotism=Y 3 ==> United States=Y Security=Y Indian-Tribes=Y 3 conf:(1)

Rule Set 2: 1860-1880

1. Republic=Y 3 ==> Congress=Y Treaty=Y Native=Y Indian=Y 3 conf:(1)
2. Congress=Y 3 ==> Republic=Y Treaty=Y Native=Y Indian=Y 3 conf:(1)
3. Treaty=Y 3 ==> Republic=Y Congress=Y Native=Y Indian=Y 3 conf:(1)
4. Native=Y 3 ==> Republic=Y Congress=Y Treaty=Y Indian=Y 3 conf:(1)
5. Indian=Y 3 ==> Republic=Y Congress=Y Treaty=Y Native=Y 3 conf:(1)
6. Republic=Y Congress=Y 3 ==> Treaty=Y Native=Y Indian=Y 3 conf:(1)
7. Republic=Y Treaty=Y 3 ==> Congress=Y Native=Y Indian=Y 3 conf:(1)
8. Republic=Y Native=Y 3 ==> Congress=Y Treaty=Y Indian=Y 3 conf:(1)
9. Republic=Y Indian=Y 3 ==> Congress=Y Treaty=Y Native=Y 3 conf:(1)
10. Congress=Y Treaty=Y 3 ==> Republic=Y Native=Y Indian=Y 3 conf:(1)

DATABASE OF SITACs AND RANKING OF RULES

We created a database storing pairs of SITACs along with their priority. Figure 4 shows the structure of the SITAC database with a few sample entries. The database and schema were developed using Oracle. We stored the SITACs as pairs ranked according to their priority such that 1 is the highest priority. In figure 4 , NAME1 and NAME2 denote the SITACs, TIMESTAMP1 and TIMESTAMP2 denote their respective timestamps and IMPORTANCE denotes their priority. Note that we have shown a selective sample here to capture rules with different priorities.

Originally, all the Sitacs were stored with the TIMESTAMP of their respective documents (Figure 4) in a Sitac_timestamp table.

NAME1	TIMESTAMP1	NAME2	TIMESTAMP2	IMPORTANCE
Union	1802	United-States	1978	1
Government	1802	Constitution	1978	1
Union	1830	United-States	1942	1
Union	1810	United-States	1965	1
Tribes-of-Indians	1802	Native-Americans	1909	1
Congress	1940	Senate	1810	1
Treasury	1810	Finance	1930	1
Treasury	1790	Finance	1921	1
British-Isles	1819	United-Kingdom	1983	1
British-Isles	1849	United-Kingdom	1958	1
British-Isles	1854	United-Kingdom	1976	1
Tribes-of-Indians	1802	Native-Americans	1908	1
Colonies	1874	Union	1854	2
Patriotism	1827	United-States	1865	2
Colonies	1841	Union	1814	2
Patriotism	1882	United-States	1834	2
Patriotism	1852	United-States	1974	2
Patriotism	1892	United-States	1974	2
Supreme-Court	1812	America	1874	3
Republic	1812	Congress	1865	3
Republic	1854	Congress	1874	3
Republic	1898	Congress	1834	3
settlement	1871	England	1884	3
settlement	1845	England	1814	3
America	1891	Rail	1884	3
America	1843	Rail	1862	3

Figure 4: SITAC table with Timestamp

In addition, the SITAC table without timestamps is also shown in Figure 5. In this figure the columns are NAME1, NAME2 and importance. This table helps us to see just the correlation of the concepts.

```
SQL> select * from sitacs order by 3 asc;

NAME1                          NAME2                  IMPORTANCE
------------------------------ ---------------------- ----------
Union                          United-States                   1
Tribes-of-Indians              Native-Americans                1
Republic                       Congress                        1
Liberty                        America                         1
Peace-Treaty                   Treaty-of-Paris                 1
Veterans                       Vietnam-War-Veterans            1
Great-Britain                  United-Kingdom                  1
Articles-of-Confederation      Constitution                    1
Bill-of-Rights                 Amendment                       1
War                            Civil-War                       1
Colonies                       Union                           1
Congress                       Senate                          1
British-Isles                  United-Kingdom                  1
Treasury                       Finance                         1
America                        Freedom                         2
Patriotism                     United-States                   2
Government                     Constitution                    2
America                        World                           2
Investment                     Union                           2
Independence                   England                         2
Liberty                        Rail                            3
Court                          America                         3
settlement                     England                         3
Legislation                    settlement                      3
America                        Rail                            3
Indian                         Republic                        3
```

Figure 5: SITAC Table with Sample Entries Only

3.1 Ranking of Rules -- Discussion on Evaluation

This section explains the method to obtain the above results (Figure 4 and Figure 5). After getting all the RULES using WEKA, we stored them in flat file with timestamps. Since we have considered the Presence and Absence of concepts in a document instead of frequency, we had to manually calculate the frequency of concepts in the documents. Because of this constraint we had to perform rest of the experiments (Ranking of Rule) in a small set of data.

We have used Cosine Similarity measure to rank the rules [6]. Cosine similarity is a measure of similarity between two vectors of n dimensions by finding the cosine of the angle between them, often used to compare documents in text mining. Given two vectors of attributes, A and B, the cosine similarity, θ, is represented using a dot product and magnitude as

Similarity = Cos(Θ) = A.B / ||A||.||B||

The cosine similarity can be seen as a method of normalizing document length during comparison.

The resulting similarity ranges from -1 meaning exactly opposite, to 1 meaning exactly the same, with 0 indicating independence

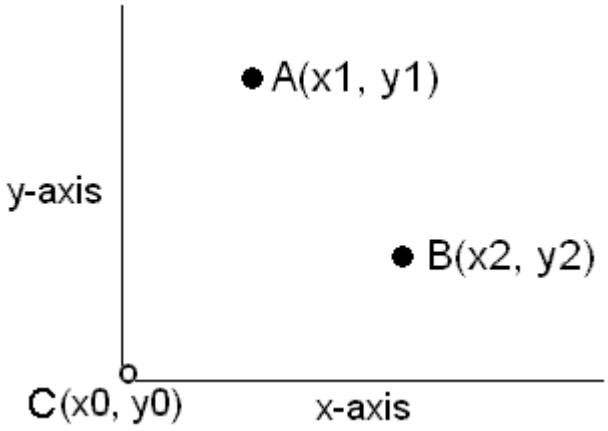


Figure 6: Cosine graph

If we multiply the coordinates of A and B and add the products together we get the "mythical" DOT Product, also known as the inner product and scalar product. So the A•B DOT Product is given by

Equation 1: $A \bullet B = x1*x2 + y1*y2$

The magnitude of a vector x of real numbers in a Euclidean *n*-space is most often the Euclidean norm, derived from Euclidean distance: the square root of the dot product of the vector with itself:

where **x** = [x1, x2, ..., xn]. For instance, the magnitude of [4, 5, 6] is $\sqrt{(4^2 + 5^2 + 6^2)} = \sqrt{77}$ or about 8.775.

Since we have considered the presence and absence of a concept in a document (instead of frequency count of a concept), we manually configure the frequency of a concept in a time period of every 10 years. So for us the starting point of the above graph was C(1790,0) and X-axis measures Years, every co-ordinate of X-axis was equivalent to 10

years. Y-axis measures the Frequency count of a concept in a document. In our experiment, the graph looks like the following picture:

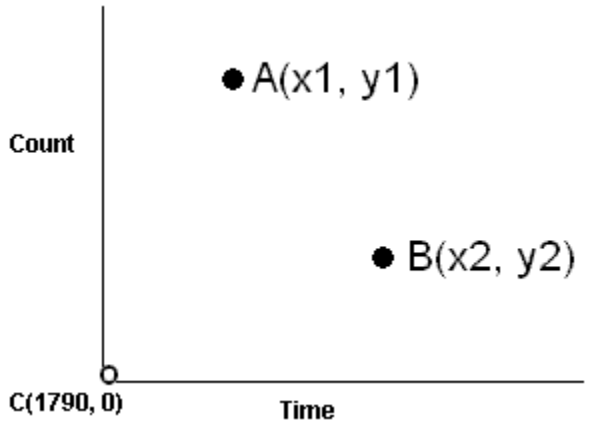


Figure 7: Cosine graph with Time and Frequency Count

To use the Cosine similarity measures we have developed a PL/SQL code segment. We have created an external table in the SITAC database. This external table can read the flat file and store the data from flat file to database. The PL/SQL code reads all the data from the external table using a PL/SQL CURSOR. For every record in the CURSOR it starts a LOOP, does the COSINE calculations and inserts the record in the external table. So at the end we have an external table with all the vectors (Pair of Concepts) and their respective Cosine values.

We have created External table in the database using below code (Figure 8).The external table definition is an Advance feature in Oracle database. It is a link between an external flat file (Plain ASCII format) and oracle database. This is a very efficient way to load the data in the database from an external source. We have taken the advantage of Oracle External Table.

```
create table ext_sitac (
  name1 varchar2(50),
  x1 number(4),
  y1 number(4),
  name2 varchar2(50),
  x2 number(4),
  y2 number(4)
  )
  organization external
  ( type oracle_loader
    default directory ext_dir
    access parameters
    ( records delimited by newline
      fields terminated by ','
    )
  location ('sitac_final.txt') )
  REJECT LIMIT UNLIMITED;
```

Figure 8: External Table code segment for Ranking of Rule

```
declare
v_numerator number;
v_denominator number;
v_cosine number;

v_name1 varchar2(50);
v_name2 varchar2(50);
v_x1 number(4);
v_y1 number(4);
v_x2 number(4);
v_y2 number(4);

cursor c_sitac is select * from ext_sitac;

begin

for rec_sitac in c_sitac
loop

  v_name1 := rec_sitac.name1;
  v_x1 := rec_sitac.x1;
  v_y1 := rec_sitac.y1;

  v_name2 := rec_sitac.name2;
  v_x2 := rec_sitac.x2;
  v_y2 := rec_sitac.y2;

  v_numerator := (v_x1 * v_x2) + (v_y1 * v_y2);
```

```
  v_denominator := sqrt((v_x1 * v_x1) + (v_x2 * v_x2)) *
sqrt((v_y1 * v_y1) + (v_y2 * v_y2));
  v_cosine := v_numerator / v_denominator;

  insert into sitac_cosine values (v_name1, v_name2,
v_cosine);
end loop;
commit;
exception
  when others then null;
end;
/
```

Figure 9: PL/SQL code segment for Ranking of Rule

We have used following query to get the Concept Pairs sorted according to their COSINE values.

```
SQL> select * from sitac_cosine order by cosine desc;

NAME1                      NAME2                         COSINE
-------------------------- ------------------------- ----------
Peace-Treaty               Treaty-of-Paris           .999957955
Colonies                   Union                     .999947852
War                        Civil-War                 .999937762
British-Isles              United-Kingdom            .999937142
Articles-of-Confederation Constitution               .999927967
Legislation                settlement                .999917959
Great-Britain              United-Kingdom            .998617829
Liberty                    America                   .998460353
Republic                   Congress                  .981829475
Patriotism                 United-States             .970879659
Congress                   Senate                    .895118173
Treasury                   Finance                   .891978673
Bill-of-Rights             Amendment                 .849025697
Tribes-of-Indians          Native-Americans          .782623792
Veterans                   Vietnam-War-Veterans      .744652494
America                    Freedom                   .742480567
Union                      United-States             .626946035
Government                 Constitution              .612173193
America                    World                     .528301887
Suprim-Court               America                   .488450009
Indian                     Republic                  .324324324
settlement                 England                   .273688184
Liberty                    Rail                      .227413946
America                    Rail                      .214754908
Investment                 Union                     .110769231
```

Figure 10: Partial output of Ranking of Rule

After getting the above result, we have created another table SITACS to store the importance of all the Concepts vectors. Importance is calculated based upon the COSINE result of Sitac_Cosine table. Cosine 0 means independent, Cosine 1 implies exactly same concepts.

Among all the association rules obtained in our experiments, we summarize the 12 best rules here with reference to context. These are the most interesting rules that represent the temporal relationships between the corresponding concepts, i.e., the rules that mine the most useful SITACs. The time-stamps though not shown here, are implicit from the corresponding documents. Hence the rules and corresponding SITACs serve the purpose of forming the basis for time-aware query translation to capture temporal terminology evolution.

Best Association Rules Obtained to Discover SITACs

1. (Union) => (United States)
2. (Tribes of Indians) => (Native Americans)
3. (Treasury) => (Finance)
4. (British Isles) => (United Kingdom)
5. (Congress) => (Senate)
6. (Colonies) => (Union)
7. (War) => (Civil War)
8. (Bill of Rights) => (Amendment)
9. (Articles of Confederation) => (Constitution)
10. (Great Britain) => (United Kingdom)
11. (Veterans) => (Vietnam War Veterans)
12. (Peace Treaty) => (Treaty of Paris)

An important observation is that these rules do not always represent synonymous terms, e.g., when President Lincoln referred to the War, he was talking about the American Civil

War as we know it today and when the presidents in the 20th century refer to it, they explicitly use the term Civil War. Likewise, when presidents Washington and Jefferson referred to the Colonies, they meant the original thirteen colonies that formed the United States at the time of independence. Later names used were the Union and finally the United States.

Properties of Rules. From all our experiments, it has been observed that the association rules discovering the SITACs follow the commutative and transitive properties in the literature. These are discussed below.

- *Commutative Property:* For any given rule, if *A=>B* automatically means *B=>A,* the rule is said to be commutative. Our discovered SITACs are found to satisfy this property.
- *Transitive Property:* For any rules, if *A=>B* and *B=>C* leads to the rule that *A=>C,* the rules are considered to be transitive. It is found that our discovered SITACs also satisfy the transitive property.

However, it is to be noted that these are the observations based on the experiments conducted so far. In the long run more evaluation will be with more text corpora, as ongoing work in this large project. Then these properties can be further addressed.

The SITACs discovered from the text sources depict the temporal relationships between the concepts and hence set the stage for time-aware query translation over the text archives online. Using the SITACs to perform the query translation forms part of the ongoing work in the whole project.

RELATED WORK

The mining of sequential patterns has been studied in the literature. Agrawal et al. in [3] they propose two algorithms, AprioriSome and AprioriAll, for mining a large database of customer transactions to discover association rules over sequences. In [14], the authors enhance their earlier work presenting an algorithm called GSP for discovering generalized sequential patterns. It is faster than the AprioriAll algorithm.

A similarity measure for structural context called SimRank [8] has been proposed in. SimRank uses a graph theoretic model where two objects are considered to be similar if they are related to similar objects. In our work, similarity measures can be defined analogous to SimRank using a recursive formulation. They consider a bipartite case and we can extend this concept to multipartite.

Strehl et al. [15] perform a comparative study of the impact of similarity metrics on cluster quality. They compare four popular similarity measures, i.e.,, Euclidean, cosine, Pearson correlation and Jaccard. They conduct a number of experiments and use t-tests to assure statistical significance of results. Cosine and Jaccard similarities emerge as the best measures to capture human categorization behavior. Note that we have used the Jaccard coefficient for ranking of rules in our approach.

Norvag et al. [11] define temporal association rules for document collections. They have 5 types of rules: episode rules, sequence rules, trend dependencies, calendar rules and inter-transaction rules. Of particular interest to us are the intertransaction rules which deal with relations within transactions. We could, to some extent, draw an analogy between their work and ours. We consider concepts within documents where each concept could be analogous to their relation and each document to their transaction. Following their method might help us to derive rules of the type ”concept c1 at time t1 and concept c2 at time t2 => concept c3 at time t3. However, this is not exactly our goal with respect to inferring how concepts change over time. We need to derive rules of the type ”concept c1 at time t1 => concept c2 at time t2”. Moreover, they do not address the use of such rules in query processing. In our work, users pose queries based on which we have to either infer such rules on-the-fly or pre-compute and store them, accordingly answer the queries and rank the responses.

Hasegawa et al. [4] define similarity and association rules by tagging named entities and getting co-occurrence pairs, and then measuring the context similarity for clustering and labeling. They have used the context vectors and frequency threshold to measure the similarity. The approach is somewhat different in our case. Our concepts are also frequency-based but in addition have to cater to the time-aware translation of queries.

In [12], the authors perform the mining of subsequences that are frequent using minimum support levels and extend this paradigm to sorting. Their techniques mine sequences taking into account user interaction and database updates.

CONCLUSIONS

We have addressed the issue of discovering SITACs in text archives, i.e., Semantically Altering Temporally Identical Concepts, with the broader goal of performing time-aware translation of user queries over these text archives online.
In this, paper we make the following contributions:

1. Proposing a methodology to discover the SITACs based on association rule mining, addressing the issues involved.
2. Implementing a solution using natural language processing over the text sources with experimentation over real corpora.
3. Developing a database of SITACs for future use in time-aware query translation.

Ongoing work includes experiments with text corpora on other topics, comparative studies with the state-of-the-art, further work on commutative and transitive properties of rules, minimizing human intervention in rule pruning and developing more advanced text parsing methods for mining. As stated earlier, this work is in collaboration with Max Planck Institute, Germany. Some of this also involves collaborative work with researchers from the Department of Linguistics at Montclair State University.

ACKNOWLEDGMENTS

This project could not have been prepared if not for the help and encouragement of various people. Hence, I would like to thank Dr. Aparna Varde, my Project Advisor for her constant support and help. I got proper guidance at every step of this project. All the materials, books, web sites that she provided were very appropriate for my project.

I would like to thank Dr. Anna Feldman from the Department of Linguistics at Montclair State University. Her invaluable guidance in the field of linguistic parser use has been an immense help to us.

I acknowledge Dr. Gerhard Weikum, Srikanta Bedathur and Klaus Berberich from Max Planck Institute for their precious comments during this work, especially while we authored a paper on this research. In general, I would like to thank the faculty staff and students in Department of Computer Science at Montclair State University for their co-operation.

REFERENCES

[1] Agrawal R, Imielinski T, Swami AN. "Mining Association Rules between Sets of Items in Large Databases." *SIGMOD*. June 1993, **22**(2):207-16, pdf.

[2] Agrawal R, Srikant R. "Fast Algorithms for Mining Association Rules", *VLDB*. Sep 12-15 1994, Chile, 487-99, pdf/pdf, ISBN 1-55860-153-8.

[3] Agrawal, R., Srikant, R.: ”Mining Sequential Patterns”. In proceedings of ICDE (March 1995), Taipei, Taiwan, pp. 3–14.

[4] Hasegawa, T., Sekine S. and Grishman R., “Discovering Relations among Named Entities from Large Corpora”, In proceedings of ACL (2004), pp. 415-422.

[5] http://en.wikipedia.org/wiki/Association_rule_learning

[6] http://en.wikipedia.org/wiki/Cosine_similarity.

[7] Ian H. Witten (Author), Eibe Frank (Author): "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations"

[8] Jeh., G. and Widom., J.: ”SimRank: A Measure of Structural-Context Similarity”. In proceedings of KDD (July 2002), Edmonton, Alberta, Canada, pp. 538–543.

[9] Jiawei Han, Micheline Kamber: “Data Mining Concepts and Techniques.”

[10] Lesh, N., Zaki, M.J. and Ogihara, M.: ”Mining Features for Sequence Classification”. In proceedings of K DD (August 1999), San Diego, California, pp. 342 – 346.

[11] Norvag, K., Eriksen, T.O. and Skogstad, K.I : ”Mining Association Rules in Temporal Document Collections”, Technical Report, Department of Computer and Information Systems (2006), NTNU, Norway.
[12] Parthasarathy, S., Zaki, M.J., Ogihara, M., Dwarkadas, S.:”Incremental and Interactive Sequence Mining”.In proceedings of CIKM (November 1999), Kansas City, Missouri, pp. 251–258.
[13] Roychoudhury D. and Varde A., “Termnilogy Evolution in Web and Text Mining Using Association Rules”, Department of Computer Science, Montclair State University, Montclair, NJ.
[14] Srikant, R. and Agrawal, R.: ”Mining Sequential Patterns: Generalizations and Performance Improvements”. In proceedings of EDBT (Mar 1996), Avignon, France, pp. 3–17.
[15] Strehl A. Ghosh, J. and Mooney R.: “Impact of Similarity Measures on Web-page Clustering”, In proceedings of AAAI, (July 2000), pp. 58-64.
[16] ”U.S. Presidential Inaugural Addresses”. In The Project Gutenberg EBook of U.S. Presidential Inaugural Addresses, www.gutenberg.net (Jan 2004), EBook Number 4938, Edition 11.