Time Aware Query Expansion over Text Archives

Aparna Varde, Srikanta Bedathur and Gerhard Weikum

Databases and Information Systems
Max Planck Institute for Informatics
Saarbruecken, Germany
August 2008

Abstract. In this research we consider text documents and address the issue of time aware query expansion. When text documents are archived, the terminology in them could undergo signficant changes over time. Thus, when a user query is expanded, it should be intelligent enough to incorporate this evolving terminology. For example, a query should be able to take into account the fact that the terms *Beijing* and *Peking* refer to the same concept at different time periods. Besides places, this logic also applies to other terms such as names that may vary over time, e.g., *Mother Teresa* whose birth name was *Anjeze Gonxhe Bojaxhiu.* Our goal is to automatically infer the congruence between such semantically identical concepts in text corpora while responding to user queries. In order to address this, we propose a solution approach that is inspired by classical concepts in association rule mining. We present herewith our problem definition, motivating examples, proposed approach, details of methodology, experimental plan and related work. To the best of our knowleddge, ours is among the first works to conduct research in the area of time aware query expansion to incorporate terminology evolution, and our solution based on association rule mining is rather unique.

1 Introduction

We have a stream of time-stamped documents in the form of newswire articles, blog posts, web-pages etc. Each document in such a setting can be seen to have a *begin-time* and can potentially have an *end-time* – together these two timestamps delimit the *validity time* of the document. These timestamped documents are archived and provided for search-ranking functionality over them.

When these document archives cover long spans of time, the terminology within these documents could undergo significant changes. This could be due to the events that trigger incorporation or dropping of terms, or due to the change in the language usage patterns etc.

As a consequence of this terminology evolution, users are faced with difficult task of carefully formulating their queries such that documents that are relevant for their information need are returned, even if the relevant document does not exactly match the query – but matches the terms in the query *prior to their evolution.*

We cast this as an extension of query expansion paradigm, wherein a user specified query over the slice of document archive (typically *now*), we include additional terms into the query such that the expanded query ranks relevant documents higher than the user-specified query. Note that, due to the presence of temporal evolution in terminology, this query expansion has to be *time-aware.*

1.1 Motivating Examples

In order to emphasize the importance of time-aware query translation, we present some examples from real data. The text source for these examples is the Gutenburg corpus with the Presidential Inaugural addresses of the USA [7]. A few sample queries over this corpus are as follows:

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?

Answers to queries 1 through 4 can be found in Abraham Lincoln's presidential address on March 4, 1861 [7]. However, he refers to the "USA" as the "Union".

Query 5 can be answered from the speeches of several presidents [7]. They use different names for the "British Isles" although they are basically referring to the same entity. We typically call it the "United Kingdom" today. James Madison in 1813 uses the term "British Isles", John Quincy Adams in 1825 says "Great Britain" while William Henry Harrison in 1841 calls it "England".

The answer to Query 6 can also be found from multiple speeches [7]. For example, James Monroe in 1817 and again in 1821 refers to the nation's policy on "Indian tribes" which we know today as "Native Americans".

Note that this is not just an issue of synonymy that we are trying to address. For instance, the terms "England" and "British Isles" are not really synonymous. However, in the given context, their implication is identical. When the USA Presidents are talking about their relations with England or with the British Isles, they are semantically addressing the same concept at different periods of time. The terms "USA" and the "Union" would also not be detected in the literature as synonyms. However, from the knowledge of a history of the United States, it is known that when the former presidents referred to the "Union", they were really talking about the "USA" as we call it today. The terms "Native Americans" and "Indian tribes" would probably be identified as synonyms. A crucial issue here however is to avoid confusion between the term "Indian" today more commonly referring to citizens of India, who could be residing in USA. Consequently "Indian tribes" today could also mean the tribal inhabitants of

India and may be referred to in issues on the USA foreign policy with developing countries. Thus, it is important to know the precise meaning of a term with reference to context and take into account how the names of the terms change over time.

Our aim is thus to be able to automatically infer the congruence between such semantically identical concepts in text corpora while responding to user queries. Based on the above example, we now formulate a definition of the problem we address in this paper.

1.2 Problem Definition

We are given text sources of time-stamped documents. They contain terms whose names may vary over time although they are in principle addressing the same concept. In our work we refer to such a concept as a *SITAC*. This is an acronym for a "Semantically Identical Temporally Altering Concept". Thus, a *SITAC* is defined as a term (word or phrase) that represents the same concept in theory but has different names in practice with respect to the time factor.

In our problem, we address the following three goals:

1. Given text corpora with time-stamps, identify the concepts whose names change over time, i.e., automatically infer the "SITACs".
2. Answer the user queries taking into account such concepts, thus making the query translation "time-aware".
3. Rank the responses incorporating the domain semantics and the temporal factors.

Having thus defined the problem, we now discuss our proposed solution based on association rule mining. The rest of this report is organized as follows. Section 2 gives the details of our solution approach consisting of three steps, namely, extraction of triplets having documents, concepts and frequencies; derivation of associations; and ranking of rules. Section 3 gives a brief outline of the experimentation plan. Section 4 surveys related work in the area. Section 5 states the conclusions along with our ongoing work.

2 Proposed Solution: Association Rule Mining Approach

Consider that the text sources are archived such that we have the information available in the form of documents and concepts (terms) over time. In other words, we have the data stored in 3 dimensions:

- Document: This is the actual text source that contains the concepts. Each document is denoted as D and has a time-stamp T associated with it.
- Concept: This refers to an individual term consisting of a word or a phrase. We denote each concept as C.
- Frequency: The number of times each concept occurs within a document is its frequency. Consider this to be denoted by F.

We thus have triplets consisting of time-stamped documents, concepts and their respective frequencies. Each such triplet would thus be denoted as (D, C, F).

Given this formulation, our association rule mining approach proposes to discover rules that identify the SITACs. Each rule is of the type: $(C1, T1) => (C2, T2)$. In other words, this translates to $(C1, D1) => (C2, D2)$ and can thus be inferred from the corresponding documents. Thus, if the input consists of time-stamped documents with concepts and their frequencies, the output is a set of temporal association rules indicating that concept $C1$ at time $T1$ implies concept $C2$ at time $T2$.

There are several issues to be considered in mining such association rules.

- Granularity of Time: The granularity of the time dimension is significant. It could have different levels ranging from days to years.
- Strength of Correlations: It is important to measure the strength of each correlation defined by an association rule, i.e., to what extent is it true that $(C1, T1) => (C2, T2)$.
- Commutative Property: An association rule is commutative if $A => B$ automatically leads to $B => A$. We need to explore commutativity in our context. In other words, if it is known that $(C1, T1) => (C2, T2)$, then we need to determine if the rule $(C2, T2) => (C1, T1)$ automatically holds good. This needs to be addressed with reference to the strength of the correlation.
- Transitive Property: Transitivity in association rules means that if $A => B$ and $B => C$, then $A => C$. It is interesting to find out whether the rules in our problem are transitive. Thus, the issue is to find out whether $(C1, T1) => (C2, T2)$ and $(C2, T2) => (C3, T3)$ can be used to infer that $(C1, T1) => (C3, T3)$. This again would involve a consideration of the strength of the respective correlations.

The traditional interestingness measures in association rule mining, namely, rule confidence and rule support could potentially be useful in measuring the strength of the correlations and in deducing whether the transitive and commutative properties apply. However, this requires deeper analysis.

We now describes the methodology used in mining temporal association rules to discover SITACs.

2.1 Details of Methodology

The process of discovering the association rules has three main steps:

1. Extraction of Triplets: Find (D,C,F) triplets from the text corpus
2. Derivation of Associations: Infer rules of the type $(C1, T1) => (C2, T2)$
3. Ranking of Rules: Arrange the rules from the most to the least interesting

These steps are explained in detail below providing equations whereover necessary. A basic introduction of text mining principles is assumed and is not overviewed in this explanation. Also note that the pruning of uninteresting rules can be an optional fourth step here. However, we consider it as an optional part of the third step dealing with post-processing and briefly discuss it accordingly.

STEP 1: EXTRACTION OF TRIPLETS

Consider the following examples with reference to the text corpus of the USA Presidential Inaugural Addresses. With reference to the sample queries described earlier, we have the following examples of time-stamped documents along with the concerned concepts at the given values of time in years.

1. – Document with Time-stamp: Abraham Lincoln's Speech, 1861
   – Concept: The Union
   – Frequency: 9
2. – Document with Time-stamp: John Quincy Adams' Speech, 1825
   – Concept: Great Britain
   – Frequency: 4
3. – Document with Time-stamp: James Madison's Speech, 1813
   – Concept: British Isles
   – Frequency: 6
4. – Document with Time-stamp: William Henry Harrison's Speech, 1841
   – Concept: England
   – Frequency: 3
5. – Document with Time-stamp: James Monroe's 1st Speech, 1817
   – Concept: Indian Tribes
   – Frequency: 5
6. – Document with Time-stamp: James Monroe's 2nd Speech, 1821
   – Concept: Indian Tribes
   – Frequency: 2

By examining such data, we can extract (D,C,F) triplets. Note that, since all the documents in these examples are from the text corpus of Presidents' speeches we abbreviate each document using just the last name of the President for convenience. We thus have the following triplets:

1. (Lincoln, The Union, 9)
2. (Quincy Adams, Great Britain, 4)
3. (Madison, British Isles, 6)
4. (Harrison, England, 3)
5. (Monroe, Indian Tribes, 5)
6. (Monroe, Indian Tribes, 2)

Thus, the process of extraction of the triplets involves first anticipating typical user queries over the source corpus. This is followed by determining the terms consisting of words / phrases in the queries that form the concepts. In other words this involves removing stop words within the queries and retaining the other terms, using stemming and lemmatization if needed. Once the concepts are thus determined, it is then required to find documents containing those concepts. A time-stamp must be attached to each document which indicates when the concept was applicable. In some cases, the time-stamp could simply be the time of creation of the document. In this case, the time-stamp instead is the year in the which the speech was made, which is history data, i.e., metadata for the document. Another point to be noted here is the granularity of the time (days, months, years) which must be taken into account. Thereafter, a frequency count of each concept is taken and stored along with the concept. Given this overview of the process involved with triplet extraction, we now describe the next step.

STEP 2: DERIVATION OF ASSOCIATIONS

The process of rule derivation consists of examining each and every pair of concepts from the time-stamped documents and executing the Apriori Algorithm for association rule mining using given thresholds of minimum confidence and minimum support. We use the analogy of market basket data for which we first need to define a transaction.

A *transaction* in the context of our problem is denoted as X and is defined as a set of correlated documents. The correlation of documents can be defined as follows. Documents are considered to be correlated if they satisfy any one of the following criteria:

1. They are referenced in a single query, as found from query logs
2. They are contained within a cluster, where the clustering is performed using standard similarity measures
3. They co-exist at any given time point
4. They have any particular concept in common

With reference to any of these alternatives, we can preprocess the data in the form of transactions consisting of correlated documents. Consider n to be the total number of transactions and m to be the total number of documents. We then have a transaction log with documents as follows. Note that (Y/N) denotes the presence or absence of a document D within a transaction X.

$X1 : [D1, Y/N], [D2, Y/N] \ldots [Dm, Y/N]$
$X2 : [D1, Y/N], [D2, Y/N] \ldots [Dm, Y/N]$
$\ldots$
$\ldots$
$Xn : [D1, Y/N], [D2, Y/N] \ldots [Dm, Y/N]$

Using this information and the data on the (D,C,F) triplets extracted earlier, we store the data in the form of concepts and their frequencies within each document. Consider m to be the total number of documents and p to be the total number of concepts. Thus, we have the following document log with concepts, where F_{C1}^{D1} denotes the frequency of concept $C1$ within document $D1$ and so forth.

$D1 : (C1, F_{C1}^{D1}), (C2, F_{C2}^{D1}) \ldots (Cp, F_{Cp}^{D1})$
$D2 : (C1, F_{C1}^{D2}), (C2, F_{C2}^{D2}) \ldots (Cp, F_{Cp}^{D2})$
$\ldots$
$\ldots$
$Dm : (C1, F_{C1}^{Dm}), (C2, F_{C2}^{Dm}) \ldots (Cp, F_{Cp}^{Dm})$

From this information, we can construct a transaction log with concepts. Consider a simple example where transaction $X1$ consists of documents $D1$ and $D2$. Thus, the transaction log with documents would have the following entry:

$X1 : [D1, Y], [D2, Y], [D3, N] \dots [Dm, N]$

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

$X1 : [D1], [D2]$

Now based on the concepts contained within these documents, the transaction log with concepts would have the following corresponding entry:

$X1 : [(C1, F_{C1}^{D1}), (C2, F_{C2}^{D1}) \dots (Cp, F_{Cp}^{D1})], [(C1, F_{C1}^{D2}), (C2, F_{C2}^{D2}) \dots (Cp, F_{Cp}^{D2})]$

Likewise, we can construct each entry in the transaction log with concepts. This serves as the preprocessed data for applying the Apriori algorithm for association rule mining. Using suitable minSup thresholds the algorithm proceeds to find frequent, 1-itemsets, i.e., frequent single concepts within a transaction, followed by frequent 2-itemsets and so forth. It then uses the minConf thresholds to obtain rules consisting of pairs of concepts across documents within each transaction. Since each document has a time-stamp associated with it, a rule such as $(C1, D1) => (C2, D2)$ which is an outcome of this Rule Derivation step would automatically mean $(C1, T1) => (C2, T2)$ which is a temporal association rule. We can extend this further to mine associations across transactions, i.e., inter-transaction association rules.

Hence, after applying the Apriori algorithm, we would get several temporal association rules based on frequency using the minSup and minConf thresholds. It is useful to rank these in the order of importance. We now describe the process of ranking as the next step.

STEP 3: RANKING OF RULES

The temporal association rules derived from the text corpora are ranked in descending order of importance, i.e., from the most to the least interesting. This ranking is performed based on an interestingness measure called a correlation score for concepts.

A correlation score S for any two concepts $C1$ and $C2$ is defined as the strength of the correlation with reference to context. It takes into account the fundamental similarity between the concepts and their frequency of co-occurrence. This is denoted as $S(C1, C2)$ read as "correlation score of concepts $C1$ and $C2$" and is calculated as follows.

The fundamental similarity between concepts $C1$ and $C2$ refers to their closeness in terms of dictionary meaning, domain semantics and such contextual factors. In other words this relates ontological definitions. Hence, we measure this similarity in terms of the distance between the concepts using standard distance measures for text. We refer to this as the ontological distance O. Consider the ontological distance between $C1$ and $C2$ denoted as $O(C1, C2)$. We now deploy the basic notion that the further the concepts are from each other with respect to distance, the less similar they are with reference to context and hence the less closely they are correlated. Thus, by the definition of correlation score and ontological distance, we find that the greater the ontological distance, the lower is the correlation score. In other words, the correlation score is inversely proportional to the ontological distance. This can be written as:

$S(C1, C2)\ \alpha\ \frac{1}{O(C1,C2)}$

Next, we consider the frequency of co-occurrence of the concepts. Our intuitive argument is that if two concepts $C1$ and $C2$ frequently co-occur in queries, they are more likely to be correlated than other concepts which are never seen together. We refer to this type of co-occurrence between concepts as concept intersection I. Hence, $I(C1, C2)$ is the concept intersection between $C1$ and $C2$ and is measured as the frequency with which concepts $C1$ and $C2$ occur together in a single query considering the aggregration as a sum over a query log. This could be a log of history data or an anticipated log of typical queries. Using this argument, we find that the greater the intersection between concepts, the higher is their correlation score. Thus, the correlation score is directly proportional to the concept intersection. Therefore, we have:

$S(C1, C2)\ \alpha\ I(C1, C2)$

Combining the two factors, namely, ontological distance and concept intersection, we get:

$S(C1, C2)\ \alpha\ \frac{I(C1,C2)}{O(C1,C2}$

This tranlates to: $S(C1, C2) = k \times \frac{I(C1,C2)}{O(C1,C2}$ where k is a constant of proportionality.

The correlation score S is used as the interestingness measure to rank the derived temporal association rules in descending order. Thus, the most interesting rule would be ranked as the topmost one based on the highest correlation score and so forth.

These ranked rules can be considered as the output of the assocaition rule mining approach to discover the *SITACs*, i.e., "Semantically Identical Temporally Altering Concepts". Hence, this ranking in principle completes the rule mining process.

However, not all the discovered rules would be interesting. Thus, it may be useful to prune the uninteresting rules as an optional post-processing step. The rule post-pruning can be done by approaches such as:

- Definition of thresholds to remove the rules that have a low rank with reference to the ranking described here
- Visual inspection based on fundamental knowledge of the domain requiring human intervention
- Integration with other methods such as the graph model, e.g., a rule is pruned if there does not exist an edge between the corresponding concepts in the graph model

The rule pruning strategies require further thought and present open issues. These, along with some of the issues identified earlier, such as transitive and commutative properties, remain a topic of our ongoing work.

It is to be noted that the association rule mining approach described here is a brute force method that would consume too much time and space to implement. Moreover, it would require huge amounts of data and metadata from text corpora in order to satisfy the minimum support and confidence thresholds. Another critique of this method is that since it takes into account only frequent concepts in the rule mining process, it may not discover rare but interesting associations between the concepts. Therefore, an important issue is to figure out whether we can modify this rule mining approach to overcome these drawbacks. Some potential suggestions include not considering all pairs of possible concepts while deriving associations but only a few top-k combinations; and using the association rule mining approach in conjunction with a other methods such as the random walk approach and / or a graph model in order to enhance performance.

3 Experimental Evaluation

3.1 Benchmark for Comparison

With reference to the examples of triplets extracted in Step 1, we find the relevant data on the concerned concepts from other sources. This relevant data includes possible alternative terms, synonyms and any other information that could come up when the concerned concept is entered in a search engine such as Google. Consider the following relevant information found from Google for the concepts in the previous examples. Note that these are only a few related concepts for each concept within the concerned documents. The related concepts are listed in random order.

- The Union:
  - The United States of America
  - State of the Union
  - European Union
  - Trade Union
  - Union of Soviet Socialist Republics
  - American Civil War Union
- British Isles:
  - United Kingdom
  - Great Britain
  - Northern Ireland
  - Ireland
  - Scotland
  - Wales
- Indian Tribes:
  - American Indian Tribes
  - Indian Affairs
  - Native American Tribes and Customs
  - American Indian Links
  - The First People of America

- American Indians
- East Indians
- India

Hence, we find the there could be many concepts that are potentially correlated to any given concept and are the outputs of queries executed over search engines. However, they may not essentially incorporate the temporal information that we seek. Therefore, the output of a search engine such as Google can serve as a benchmark for comparison with our technique in the evaluation of performance.

3.2 Discussion

The temporal association rule mining approach we have proposed for discovering SITACs would consume too much time and space to implement on a large scale. Moreover, it would need huge amounts of data and metadata from text corpora in order to satisfy the minimum support and confidence thresholds. Another critique of this method is that since it takes into account only frequent concepts in the rule mining process, it may not discover rare but interesting associations between the concepts. Therefore, an important issue is to figure out whether we can modify this rule mining approach to overcome these drawbacks. Some potential suggestions include not considering all pairs of possible concepts while deriving associations but only a few top-k combinations; and using the association rule mining approach in conjunction with a other methods such as the random walk approach and / or a graph model in order to enhance performance.

4 Related Work

A similarity measure for structural context called SimRank has been proposed in [11]. SimRank uses a graph theoretic model where two objects are considered to be similar if they are related to similar objects. For example, people are similar if they purchase similar items; and items are similar if they are purchased by similar people. Accordingly they define bipartite SimRank scores based on equations that take into account the in-neighbors and out-neighbors of nodes in graphs, where nodes represent objects and edges represent relationships between them. 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.

Clusters of keywords are discovered from blogs for specified temporal periods in [2]. They propose efficient algorithms that find correlated keywords in blogs over given time slots and identify stable clusters, i.e., those that are temporally ‘persistent. Their method involves inferring statistically significant associations among keyword-pairs in graphs and defining correlation coefficients to compute the strengths of correlations. Stable clusters are then discovered by performing a breadth first search and a depth first search over the cluster graph, using the

Threshold Algorithm [6] to find paths of certain lengths. This could potentially be relevant to our work because once the clusters are defined, terms that are members of the same cluster could be related to each other. Stable clusters could probably provide useful information with respect to identifying semantic concepts varying over time.

Social network analysis has gained tremendous popularity in recent years. In [4], dynamic social networks are analyzed based on a mathematical and computational framework that uses information about the time of occurrence of social interactions. They consider a given population, define a group to be its subset and use a set of partitions where each partition is one time step. Algorithms are outlined to identify various metagroups: most persistent metagroup, most stable metagroup and largest metagroup. The Jaccard's similarity measure is used to compute the similarity between any two groups. They identify certain critical network properties such as group connectivity, individual connectivity and critical group set. Their model can be extended to address problems such as discovering extroverts and introverts, loyal individuals and metagroup representatives. A potential relevance to our work is the identification of metagroups. Semantic concepts changing over time, in our context, may belong to the same metagroup. In [18], related issues are presented with respect to community identification in social networks. They define communities as unusually dense knit subsets of social networks. They make certain assumptions about the behavior of individuals within a community and develop a framework for detecting communities that change over time. Comparing this with our problem, the identification of users belonging to specific communities could serve to give a hint that they are related, e.g., they could even be the same user with different names. Top-k querying is performed over social networks in [16]. They propose an incremental top-k querying algorithm for social search and ranking that performs social expansion based on the strength of relations between users and semantic expansion based on relatedness among various tags. They define relations between nodes of the same type as friendship, tag similarity and linkage and relations between nodes of different types as document content, tagging and rating. A social scoring model is developed based on friendship similarity and social frequency and is used for query processing with and without tag expansion. The social frequency as defined here has some relevance to our work in terms of revealing temporal relationships. The relations they define among nodes of the same type and different types may potentially remain consistent over time, thus leading towards the same concept semantically. Thus, some of the ideas in this work could possibly be useful to us.

The LiWA project on Living Web Archives includes a description of issues such as homonomy and synonymy [17]. They find word relationships from the appearance of graphs. If two otherwise unconnected subgraphs get connected by one node, it implies homonymy, i.e., one word having multiple meanings. Pairs of nodes whether directly linked or not but strongly connected through shared neighbors indicate synonymy. In LiWA, terminology evolution is considered as well [10]. This includes neologisms, changes in word etymology and political renames. They use terminology graphs obtained from collections having the same domain and different time stamps. Exploratory graph analysis with term sampling using word statistics is also applied. In [3], a time machine for text search is proposed. They extend the inverted file index [20] to adapt it for temporal searches, perform temporal coalescing to prevent an oversized index while yet meeting the requirements of accuracy and design methods for sublist materialization to provide a trade-off between space and performance. They introduce a time travel inverted file index which in addition to containing information on the document identifier and the payload, also stores the validity of the time interval. This index could probably be useful in our problem for searching concepts that are conceptually the same but vary in name temporally.

An algorithm called TriCluster [19] has been proposed to mine clusters in three dimensions over gene expression data. TriCluster uses a graph-based approach for mining a matrix of genes and samples over a time slice. It can thus obtain temporal clusters along the gene-sample-time dimensions. It uses an approach of building a range multigraph, mining maximal biclusters from it, using that to derive maximal triclusters and optimally deleting or merging overlapping clusters. Though this approach is useful in finding significant clusters in mircoarray data, it does not seem to be directly applicable in our context. We are interested in detecting changes in concepts over time and answering user queries accordingly where rules are likely be more helpful for pinpointing which concepts have changed. In [14], time-series rule discovery is performed on gene expression data. They aim to find dependencies between genes of the type: "if gene A is active then gene B becomes active or inactive within a certain time". Their strategy is based on association rule mining using the Aprori algorithm and they use a Boolean network model to represent genetic interactions. Using DNA microarray data where a row contains expression levels of one gene at different time points and a column contains expression levels of all genes at one time point, they derive association rules with a time offset to discover activations and inhibitions between genes in a time series of expansion profiles. They modify the definitions of rule confidence and support as needed for their time-stamped association rules. Finding such dependencies between genes assists in understanding causal structures and the genetic functions of the cells. This could bear some similarity with our work. However, they try to find cause-effect relationships while we try to explore terminology evolution. In our case, words and phrases change over time, so reference to context is important. Accordingly, we need to exploit the corpus with terms and documents. Our association rule consider the time domain and also have a contextual significance.

Norvag et al. [9] define temporal association rules for document collections. They have 5 types of rules: episode rules, sequence rules, trend dependencies, calendar rules and intertransaction rules that capture different kinds of temporal relationships within documents. Of particular interest to us are the intertransaction rules which deal with relations within transactions. An example of such as rule is "car at time t0 and hotel at time t1 => leasing at time t4". 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 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 infer such rules on-the-fly or precompute them, accordingly answer the queries and rank the responses.

The mining of sequential patterns has been studied in the literature. Agrawal et al. in [1] they propose two algorithms, AprioriSome and AprioriAll for mining a large database of customer transactions to discover associations rules over sequences. They find the maximal sequences among all sequences that have a certain user-specified minimum support. In [15], the authors enhance their earlier work presenting an algorithm called GSP for discovering generalized sequential patterns. It is faster than the AprioriAll algorithm. It also considers time constraints specifying a minimum and maximum time period among adjacent items in a pattern. Zaki et al. in [13] perform the mining of subsequences that are frequent using minimum support levels and extend this paradigm to sorting. Their goal is to reduce input-output and computation needs in handling incremental updates to the data, while mining, since data sources could undergo changes. Their technqiues mine sequences taking into account user interaction and database updates. In [8], they also deal with sequential data. They develop sequence mining methods for selecting features to serve as an input for classification with algorithms such as Naive Bayes. The data in this work consists of examples, each represented as a sequences of events, each event having a set of predicates. Thus, their goals are quite different from ours. In all these sequence mining approaches, we can draw some analogy with our work. However, none of these consider terminology evolution where terms change over time.

The topic of connection subgraphs is presented in [5]. A connection subgraph is a small subgraph of a big graph which can best depict the relationship between two nodes. They produce approximate but good quality connection subgraphs on huge graphs in a real time environment. This paper uses an approach analogous to the classical Kirchhoff's current laws in electronic circuits. In the graph model approach in our work also, we can tangentially relate to this method. In [12] they propose an approach called SPIRIT for streaming pattern discovery in multiple time series. They consider several numerical data streams and incremenatlly find correlations and hidden variables to represent the main trends in the data streams. This can be considered remotelt related to the association rule mining approach in our work if we wish to perform the mining as and when user queries occur. However, we consider it more useful to precompute and materialize based on anticapted user queries and perform the mining accordingly deriving temporal association rules that are interesting. It is to be noted that in some way our work can be considered orthogonal to such literature.

5 Conclusions and Ongoing Work

In this research, we consider the problem of evolving terminology in text archives while expanding user queries on them. We consider time-stamped text documents as inputs, and propose a solution to this problem that has its roots on association rule mining. The contributions of this work are as follows.

- It is among the first of its kind to addres terminology evolution in text documents for query expansion.
- It presents interesting real-world examples to motivate the readers anout the signifance of the problem.
- It proposes a solution approach that entails association rule mining concepts and is certainly unique in that respect.
- It provides a thorough survey of the literature in the area to emphasize where our work stands out.
- It explains the details of the methodology in the solution approach considering aspects such as ranking.

Ongoing work includes conducting exhaustive experimentation using real data from online text archives, adhering to the methodology and benchmarks here. We would address further challenges as the work progresses. This research would be useful to the web data management and text mining communities, in particular.

References

1. Agrawal, R., and Srikant, R.: "Mining Sequential Patterns". In proceedings of ICDE (March 1995), Taipei, Taiwan, pp. 3–14.
2. Bansal, N., Chiang, F., Koudas, N. and Tompa, F. W.: "Seeking Stable Clusters in the Blogosphere". In proceedings of VLDB (September 2007), Vienna, Austria, pp. 806–816.
3. Berberich., K., Bedathur, S., Neumann, T. and Weikum, G.: "A Time Machine for Text Search". In proceedings of SIGIR (July 2007), Amsterdam, Netherlands.
4. Berger-Wolf, T. Y. and Saia, J.: "A Framework for Analysis of Dynamic Social Networks". In proceedings of KDD (August 2006), Chicago, Illinois, pp. 523–528.
5. Faloutsos, F., McCurley, K. S. and Tomkins, A.: "Fast Discovery of Connection Subgraphs". In proceedings of KDD (August 2004), Seattle, Washington, pp. 118–127.
6. Fagin, R., Lotem, A. and Naor, M.: "Optimal Aggregation Algorithms for Middleware". In proceedings of PODS (June 2001), Santa Barbara, California.
7. "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.
8. Lesh, N., Zaki, M.J. and Ogihara, M.: "Mining Features for Sequence Classification". In proceedings of KDD (August 1999), San Diego, California, pp. 342 – 346.
9. 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.

10. Iofeiu, T.: ”LiWA - Living Web Archives: Process Overview and Data Sources”. L3S Research Center Presentation (May 2008), Hannover, Germany.
11. Jeh., G. and Widom., J.: ”SimRank: A Measure of Structural-Context Similarity”. In proceedings of KDD (July 2002), Edmonton, Alberta, Canada, pp. 538–543.
12. Papadimitriou, S., Sun, J. and Faloutsos, C.: ”Streaming Pattern Discovery in Multiple Time Series”. In proceedings of VLDB (August 2005), Trondheim, Norway, pp. 697–708.
13. 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.
14. Schu I.: ”Time Series Rule Discovery on Gene Expression Data”. Masters Thesis, Max Planck Institute for Informatics (March 2006), Saarbrucken, Germany.
15. Srikant, R. and Agrawal, R.: ”Mining Sequential Patterns: Generalizations and Performance Improvements”. In proceedings of EDBT (Mar 1996), Avignon, France, pp. 3–17.
16. Schenkel, R., Crecelius, T., Kacimi, M., Michel, S., Neumann, T., Xavier Parreira J. and Weikum, G.: ”Efficient Top-k Querying over Social-Tagging Networks”. In proceedings of SIGIR (July 2008), Singapore.
17. Tahmasebi, N.: ”LiWA - Living Web Archives: Problem Statement and Definitions”. L3S Research Center Presentation (May 2008), Hannover, Germany.
18. Tantipathananandh, C., Berger-Wolf, T. and Kempe, D.: ”A Framework for Community Identification in Dynamic Social Networks”. In proceedings of KDD (August 2007), San Francisco, California, pp. 717–725.
19. Zhao, L. and Zaki., M.J.: ”TriCluster: An Effective Algorithm for Mining Coherent Clusters in 3D Microarray Data”. In proceedings of SIGMOD (June 2005), Baltimore, Maryland, pp. 695 - 705.
20. Zodel, J. and Moffat, A.: ”Inverted Files for Text Search Engines”. In ACM Computing Surveys (2006), Vol. 38, No. 2, Article 6.