Trend Graphs: Visualizing the Evolution of Concept Relationships in Large Document Collections

Ronen Feldman, Yonatan Aumann, Amir Zilberstein, Yaron Ben-Yehuda

Department of Mathematics and Computer Science
Bar Ilan University
Ramat-Gan, 52900 Israel
{feldman,aumann,zilbers,benyehdy}@cs.biu.ac.il

Abstract. The proliferation of digitally available textual data necessitates automatic tools for analyzing large textual collections. Thus, in analogy to *data mining* for structured databases, *text mining* is defined for textual collections. A central tool in text mining is the analysis of *concept relationship,* which discovers connections between different concepts, as reflected in the corpus. Most previous work on text mining in general, and concept relationship in particular, viewed the entire corpus as one monolithic entity. However, large corpuses are often composed of documents with different characteristics. Most importantly, documents are often tagged with *timestamps* (e.g. news articles), and thus represent the state of the domain in different time periods. In this paper we introduce a new technique for analyzing and visualizing differences and similarities in the concept relationships, as they are reflected in different segments of the corpus. Focusing on the case of timestamped documents, we introduce *Trend Graphs*, which provide a graphical tool for analyzing and visualizing the dynamic changes in concept relationships over time. *Trend Graphs* thus provide a tool for tracking the evaluation of the corpus over time, highlighting trends and discontinuities.

1. Introduction

Most informal definitions [4] introduce *knowledge discovery in databases* (KDD) as the extraction of useful information from databases by large-scale search for interesting patterns. The vast majority of existing KDD applications and methods deal with structured databases, for example - client data stored in a relational database, and thus exploits data organized in records structured by categorical, ordinal, and continuous variables. However, a tremendous amount of information is stored in documents that are nearly unstructured. The availability of document collections and especially of online information is rapidly growing, so that an analysis bottleneck often arises also in this area. Thus, in analogy to *data mining* for structured data, *text mining* is defined for textual data. *Text mining* is the science of extracting information from hidden patterns in large textual collections.

Text mining shares many characteristics with classical data mining, but also differs in some. Thus, it is necessary to provide special tools geared specifically to text mining. A central tool, found valuable in text mining, is the analysis of *concept relationship* [5,7], defined as follows. Large textual corpuses are most commonly composed of a collection of separate *documents* (e.g. news articles, web pages). Each document refers to a set of *concepts* (*terms*). Text mining operations consider the distribution of concepts on the *inter-document* level, seeking to discover the nature and relationships of concepts as reflected in the collection as a whole. For example, in a collection of news articles, a large number of articles on politician X and "scandal" may indicate a negative image of the character, and alert for a new public-relations campaign. Or, for another example, a growing number of articles on both company Y and product Z may indicate a shift of focus in the company's interests, a shift which should be noted by its competitors. Notice that in both of these cases, the information is not provided by any single document, but rather from the totality of the collection. Thus, *concept relationship* analysis seeks to discover the relationship between concepts, as reflected by the totally of the corpus at hand.

Most previous work on detecting concept relationships viewed the entire textual corpus as one monolithic entity. However, large corpuses are often composed of documents with different characteristics. In this case, we seek a mining tool which can aid in analyzing the similarities and differences between the different segments of the corpus. For example, news-articles are often gathered over an extended period of time, and each document is timestamped with a date. In this case, the mining process should not only provide analysis of the corpus as a whole, but also describe the temporal behavior of events, and provide an analysis of the evaluation of the concepts and their relationships, as reflected in the collection. In this paper we introduce a new technique for analyzing and visualizing differences and similarities in the concept relationships, as they are reflected in different segments of the corpus. Focusing on the temporal case, we introduce *Trend Graphs*, which provide a visual representation of the time-dependent behavior and evaluation of concepts and their inter-relationships, as reflects in the corpus. Trend Graphs provide a global overall picture of the dynamic behavior of corpus, highlighting important changes, trends, and discontinuities. Our notion of Trend Graphs complements the trend analysis technique of Lent et. al. [11], providing an overall picture of all major trends, and focusing on concept relationships, rather than sequences.

Document Explorer System. The *Trend Graphs* tool we describe here is part of an integrated text mining system, developed at Bar-Ilan University, named *Document Explorer*. The system gets as input a set of text documents, and provides a full suite of analysis and mining tools. As a first step, the system analyzes each document, and, using a special *Term Extraction* module, generates a set of *terms* representing the document. This process is performed only once for each document, and all subsequent analysis is performed upon the term-representation of the document.

Outline. The rest of this paper is organized as follows. Following, we provide a brief summary of related work. Next, in Section 2, we give some preliminary definitions and describe the notion of *Context Graphs*, which is the basis for the definition of *Trend Graphs*. In Section 3, we introduce *Trend Graphs*. In Section 4, we discuss the application of our methods to general inter-collection comparisons. We conclude in Section 5 with open research directions. The examples provided in the paper are based on application of the system to a set of 52,000 articles of Reuters Financial News from 1995-1996.

Related Work.

A data mining approach to analyzing large textual collections was presented by Feldman and Dagan [5]. Mining for association rules in texts is discussed be Feldman and Hirsh [6]. Visualization techniques for text mining are presented by Feldman et. al. [7]. Hahn and Schnattinger [8] describe the SYNDIKATE system for "deep KDD from natural language texts" which is aimed at extracting knowledge from real-world texts and assimilating them into a knowledge base.

The problem of discovering trends in textual collections was first considered by Lent, Agrawal, and Srikant [11]. They present a method for detecting and presenting trends in word phrases (where phases are taken to have a very broad meaning). Their method is based on a two-phase process. In the first phase they construct frequent *sequences* of words, using a sequential patterns mining algorithm [14]. Then, the user can query the system to obtain all phases whose trend matches a given pattern (such as "recent upward change"). Our Trends Graph tool presents a totally different approach and solution to the problem. Trend Graphs present *all* major trends in the corpus (not only of a given type), and are based on a visualization technique. The appreciation of trends emerges from their visual appearance.

Sequences and trends in structured databases are considered in many papers. Srikant and Agrawal [14] define *sequential patterns* and provide an efficient algorithm for mining such patterns. Mannila et. al. [12] define the notion of *episodes*, and provide an efficient algorithm for mining them. Frequent episodes are used to discover behavior patterns, and to generate behavior rules. Other work on knowledge discovery from sequential data includes [2,3,9] and others. Time series analysis is discussed in [1,10].

2. Context Graphs

Taxonomy. All operations are performed using a *taxonomy,* which is defined by the user (or preloaded together with the system). The taxonomy is a hierarchy of the terms generated by the term extraction module. Each node of the taxonomy corresponds to a set of terms. The taxonomy is necessary in order to allow the system

to focus on the relevant set of concepts. The *Document Explorer* system provides several tools to aid the user in creating the taxonomy.

Context Graphs. The basis for the definition of Trend Graphs is the notion of a *Context Graph.* A Context Graph is a visual representation of the relationship between a set of terms (e.g. "companies"), as reflected in the corpus with regards to a given context (e.g. "merger"). In order to specify a Context Graph the user chooses two sets of terms. The first set, called the *connection nodes*, defines the terms for which the user is interested in finding the inter-relationship. The second set, called the *context*, defines the context in which the relationship must be found. Both sets may be defined using the taxonomy. The context set can be treated either as an AND set, i.e., all terms must occur in the document, or as an OR set, i.e., at least one of the terms must occur in the document.

We now formally define a Context Graph, starting with some necessary notations:

- *Context Phrase*: For a collection of documents, D, and a set of terms, T, we denote by $D/A(T)$ the subset of documents in D that are labeled with *all* of the terms in T. Similarly, $D/O(T)$ is the subset of documents in D that are labeled with *at least one* of the terms in T. $A(T)$ and $O(T)$ are called *context-phrases*.

- *Context Relationship:* For D, a collection of documents, t_1, t_2 individual terms, and C a context phrase, we denote by $R(D, t_1, t_2|C)$ the number of documents in D/C which include both t_1 and t_2. Formally, $R(D, t_1, t_2|C) = |(D/A(\{t_1,t_2\}))/C|$.

- *Context Graph:* For D, a collection of documents, T, a set of terms, C a context phrase, and threshold k, the Context Graph of D,T,C is a weighted graph $G = (T,E)$, with nodes in T and set of edges $E = \{\{ t_1, t_2\} \mid R(D, t_1, t_2 \mid C) > k \}$. For each edge $\{t_1,t_2\} \in E$, we define the *weight* of the edge, $w(\{t_1,t_2\}) = R(D, t_1, t_2 \mid C)$.

Intuitively, a context graphs is a graph where vertices represent the terms, and there is an edge between vertices terms if both terms co-occur in the given "context" sufficiently many times. Figure 1 shows the Context Graph for "companies" in the context of "merger talk" (i.e. T={"companies"}, C=A("merger talk")). The threshold was set to 7. The weights of the edges are noted alongside the edge. Nodes with no edges are not depicted.

The graph clearly exposes the main industry clusters, which are shown as disconnected components of the graph: the computer industry cluster, the broadcasting cluster, a banking cluster, and a telephony cluster. The Context Graph provides a powerful way to visualize relationship encapsulated in thousands of documents.

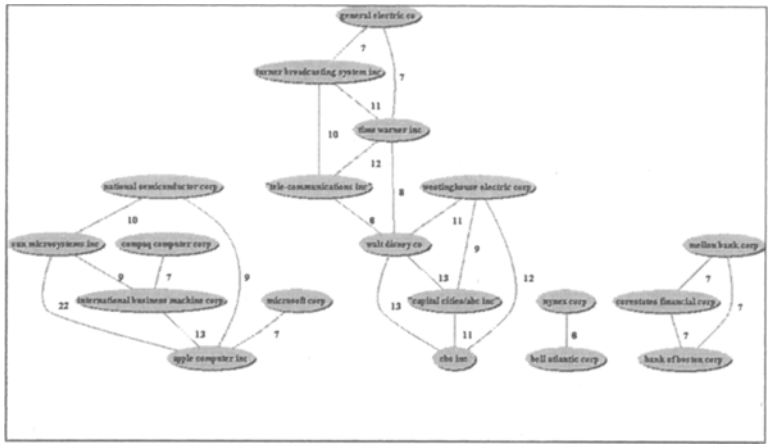

Fig. 1. Context Graph – Companies in the context of merger.

3. Trend Graphs

Each document is assumed to be tagged with a time-stamp, which indicates its creation time. For example, in the running example for this paper, each Reuters' document is tagged by its publication date.

Context Graphs, as defined in the previous section, provide a static visualization of relationship of terms in the collection as a whole. We now show how to go beyond the notion of Context Graphs, and provide a tool that tracks the *evolution* and *changes* of relationships over time. Thus, we obtain a tool that aids the user in finding trends and discontinuities. The user can select the time granularity, and then, using the visualization tool, see how the relationships change and evolve over time. The system automatically highlights new connections, and alert for changes.

We start with several formal definitions:

- *Temporal Selection*: If D is a collection of documents and I is a time range, D_I is the subset of documents in D whose time stamp is within I.

- *Temporal Context Relationship*: If D is a collection of documents, t_1 and t_2 are individual terms, C is a context phrase, and I is a time Interval, then $R_I(D, t_1, t_2|C)$ is the number of documents in D_I in which t_1 and t_2 co-occur in the context of C, *i.e.*, $R_I(D, t_1, t_2|C)$ is the number of D/C which include both t_1 and t_2.

- *Temporal Context Graph:* If D is a collection of documents, T is a set of terms, C is a context phrase, I is a time range, and k is a threshold, the Temporal Context Graph of D,T,C,I, is a weighted graph $G_I = (T,E_I)$, with set nodes in T, and set of edges E_I, where, $E_I=\{\{t_1,t_2\}|R_I(D, t_1, t_2 \mid C) > k\}$. For each edge $\{t_1,t_2\}\in E$, we define the *weight* of the edge, $w_I(\{t_1,t_2\})= R_I(D, t_1, t_2 \mid C)$.

Intuitively, a Temporal Context Graph is a Context Graph for a specific time slice. Given these definitions, a *Trend Graph* is obtained by partitioning the entire time span covered by the collection into a series of consecutive time intervals, and taking the corresponding sequence of Temporal Context graphs. The *changes* in edge weights reflect the changes in relationships over time. In order to emphasis the temporal effect, we mark each edge with information on the difference between the weight of the edge in the current time interval, and that of the previous interval. The edges are categorized into four classes:

1. New edges: edges that did not exist in the previous graph.
2. Increased edges: edges that have a much higher weight compared to the previous time interval.
3. Decreased edges: edges that have a much lower weight compared to the previous graph.
4. Stable edges: edges that have approximately the same weight as the corresponding edge in the previous graph.

When depicting the Trend Graph, each edge is marked as being a member of one of the classes. The notions of "much higher" and "much lower" can be defined according to the context. Currently we require a 1.25 factor increase or decrease as the minimum for a *significant* change.

Trend graphs can be visualized in several ways. We chose to visualize the category of each edge by either using a color scheme or by changing the width of the edge. The layout of the graph was chosen so as to minimize the changes in the layout of the nodes and edges. Thus, we first precompute the Context Graph for the entire period and determine its layout. This layout determines the position of all nodes and edges in all graphs. Thus, the entire sequence of Temporal Context graphs is defined, together with the class definition for each edge. The system now provides the user with several ways to view the sequence. The best way is by animation, where one graph transforms into the next, and the changes become most apparent. Another way the user can chose to view the sequence is by using a listbox, which contains all the periods (in the selected granularity). The user can then flip between periods and see a Trend graph that compare between the context graph of the current period and the average of the previous n periods.

Figure 3 shows the interface for the user to define a Trend Graph. In this case, the user chose to examine trends in the relationship between companies in the context of included the word "merger". In this case there are three such terms: "merger", "merger agreement", and "merger talk". The time granularity was chosen to be quarters.

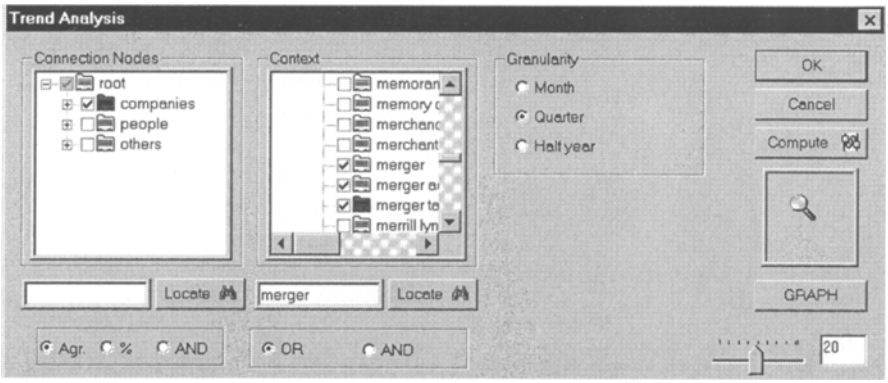

Fig. 2. Specifying a Trend Graph

Figures 4 and 5 depict the trend graphs for the forth quarter of 1995 and the first quarter of 1996. New and increased edge are depicted with a fat line, other edges are depicted with a regular line. Each edge is labeled with the difference of the edge in this graph, with regards to the average of previous quarters. A **n* label represents a new edge with weight *n*. A *>n* label represents an edge with an increase in weight. A *<n* label represent an edge with a decrease in weight.

The 1996 1st quarter graph unveils several clear trends. First, there is an overall increase in connections. More specifically, there is a sharp increase in the connection between Nynex and Bell Atlantic, and an altogether new connection between Sun and Apple. In the banking industry, the connection has decreased.

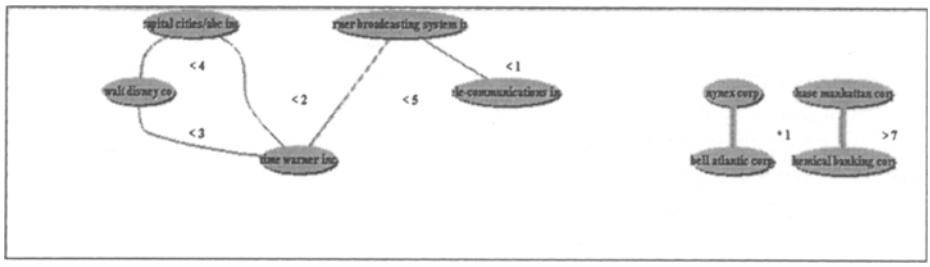

Fig. 3. Trend Graph for 4th Quarter 1995 (Companies in the context of "merger")

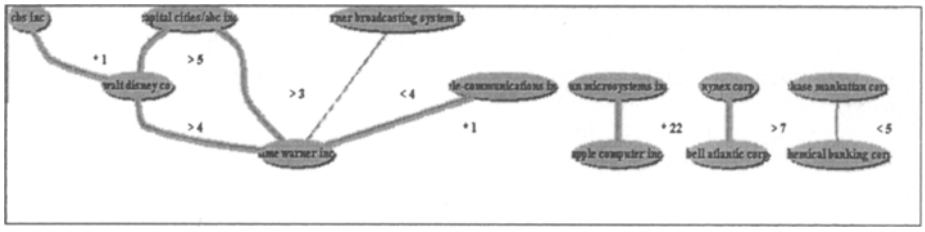

Fig. 4. Trend Graph for 1st Quarter 1996 (Companies in the context of "merger")

4. Inter-Collection Comparisons

Trend Graphs represent only one type of comparison, which can be performed using Context Graphs. In this case, the comparison is made between the documents written at different time intervals. In a similar fashion, one can use Context Graphs to find similarities and differences between any sets of document, e.g. documents coming from different sources, or different segments of the same corpus. For example, a company may wish to analyze the similarities and differences between the email message it gets from its US based customers, and its European customers. Or, one can compare the way two different newspapers reflect on a given domain. In these types of applications, the graphs shall represent the concept relationships as reflected in one collection in comparison to the other. There are several ways to present the comparison results. The first is to use the same representation used for Trend Graphs, i.e., using several graphs and labeling the edges with the differences. However, since there is no natural order of the segments, other representation may also be considered. If the comparison is only between two segments, a single graph showing the differences and similarities is possible. If there are more than two segments, one can superimpose the edges using a color-coding scheme to distinguish the segments.

Comparisons using Context Graphs can performed on the set of clusters generated by a clustering algorithm, thus obtaining a better understanding of the clusters.

5. Conclusions and Future Work

Temporal affects are central to the understanding of many systems. Thus, data mining tools, which aspire to analyze the behavior of systems, must take the time factor into account as a major component. In this paper we presented a new tool, called Trend Graphs, which captures to temporal behavior of concept relationships in large textual collections. The tool provides the user with an easy to understand graphical representation of the changes in the collection over time, highlighting trends, directions, and discontinuities.

The tools we provide can also be used to analyze differences and similarities between different segments of a given collection, or of separate collections.

An interesting research direction is the issue of the graph layout in Trend Graphs. The approach adopted in this paper is to use a static layout for all the graphs in the sequence. The layout is obtained from the union of all the graphs in the sequence. Thus, nodes are always located in the same place, and differences are depicted by varying edge weights, or edge colors. An alternative method is to have a dynamic graph layout. In this case, the relative node locations shall correspond to their relationship, e.g. closer nodes describing a stronger connection. Using this approach,

the differences between the graphs are visualized by the changes in the graph layout. In this approach, animation techniques should be used to help the user follow the changes.

References:

1. Agrawal, R.; Lin, K. I.; Sawhney, H. S.; and Shim, K.: Fast Similarity Search in the Presence of Noise, Scalling and Translation in Time-Series Databases. In: Proceeding of the Annual Symposium on Very Large DataBases (VLDB), (1995) 490-501.
2. Bettini, C.; Wang, X. S.; and Jajodia, S.: Testing Complex temporal relationships Involving Multiple Granularities and its Application to Data Mining. In: Proceedings of the 15th ACM Symposium on Principles of Database Systems (PODS), (1996) 68-78.
3. Dousson, C.; Gaborit, P.; and Ghallab, M.: Situation Recognition: Representation and Algorithms. In: Proceedings of the 13th International Joint Conference of Artificial Inteligence (IJCAI), (1993) 166-172.
4. Fayyad, U,; Piatetsky-Shapiro, G.; and Smyth P.: Knowledge Discovery and Data Mining: Towards a Unifying Framework. In: Proceedings of the 2nd International Conference of Knowledge Discovery and Data Mining (KDD), (1996) 82-88.
5. Feldman, R.; and Dagan, I.: KDT - Knowledge Discovery in Texts. In: Proceedings of the 1sr International Conference of Knowledge Discovery and Data Mining (KDD), (1995).
6. Feldman, R.; and Hirsh, H.: Mining Association Rules in Text in the Presence of Background Knowledge. In: Proceedings of the 2nd International Conference of Knowledge Discovery and Data Mining (KDD), (1996).
7. Feldman, R.; Klosgen, W.; and Zilberstein, A.: Visualization Techniques to Explore Data Mining Results for Document Collections. In: Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining (KDD), (1997) 16-23.
8. Hahn, U.; and Schnattinger, K.: Deep Knowledge Discovery from Natural Language Texts. In: Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining (KDD), (1997) 175-178.
9. Hotanen, k.; Klemettinen, M.; Mannila, H.; Ronkainon, P.; and Toivonen, H.: TASA: Telecommunication Alarm Sequence Analyzer, or "How to Enjoy Faults in Your Network". In: Proceedings of the 1996 IEEE Network Operations and Management Symposium (NOMS), (1996) 520-529.
10. Keogh, E.; and Smyth, P.: A Probabilistic Approach to Fast Pattern Matching in Time-Series Databases. In: Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining (KDD), (1997) 24-130.
11. Lent, B.; Agrawal, R.; and Srikant, R.: Discovering Trends in Text Databases. In: Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining (KDD), (1997) 227-230.
12. Mannila, H.; Toivonen, H.; and Verkamo, A.I.: Discovering Frequent Episodes in Sequences. In: Proceedings of the 1st International Conference of Knowledge Discovery and Data Mining (KDD), (1995) 210-215.
13. Mannila, H.; and Toivenen, H.: Discovering Generalized Episodes Using Minimal Occurances. In: Proceedings of the 2nd International Conference of Knowledge Discovery and Data Mining (KDD), (1996) 146-151.
14. Srikant, R.; and Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In: Proceedings of the 5th International Conference on Extending Database Technology (EDBT), (1996).