TextVis: An Integrated Visual Environment for Text Mining*

David Landau, Ronen Feldman,
Yonatan Aumann, Moshe Fresko, Yehuda Lindell,
Orly Lipshtat

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

Oren Zamir

Department of Computer Science
University of Washington
Seattle, WA
zamir@cs.washington.edu

Abstract. TextVis is a visual data mining system for document collections. Such a collection represents an application domain, and the primary goal of the system is to derive patterns that provide knowledge about this domain. Additionally, the derived patterns can be used to browse the collection. TextVis takes a multi-strategy approach to text mining, and enables defining complex analysis schemas from basic components, provided by the system. An analysis schema is constructed by dragging functional icons from a tool-pallette onto the workspace and connecting them according to the desired flow of information. The system provides a large collection of basic analysis tools, including: frequent sets, associations, concept distributions, and concept correlations. The discovered patterns are presented in a visual interface allowing the user to operate on the results, and to access the associated documents. TextVis is a complete text mining system which uses agent technology to access various online information sources, text preprocessing tools to extract relevant information from the documents, a variety of data mining algorithms, and a set of visual browsers to view the results. This paper provides an overview on the TextVis system. We describe the system's architecture, the various tools, and discuss the advantages of our visual environment for mining large document collections.

1. Introduction

Knowledge discovery in databases (KDD) is informally defined as the extraction of useful information from databases by large-scale search for interesting patterns [4]. The vast majority of existing KDD applications and methods deal with structured databases, e.g. client data stored in a relational database, and thus exploit the predefined organization of the data in the database. However, a tremendous amount of

* In: Proceedings of *The 2nd European Symposium on Principles of Data Mining and Knowledge Discovery* (PKDD98), Nantes, France. 1998.

information is also available in unstructured, textual form. The availability of document collections and especially of online information is rapidly growing. The vast amounts of information prohibit manual analysis and effective exploration. Thus, it is necessary to provide automatic tools for analyzing large textual collections. Accordingly, in analogy to *data mining* for structured data, *text mining* is defined for textual data. The goal of text mining is to extract information from patterns in large textual collections. The results can be important both for the analysis of the collection, and for providing intelligent navigation and browsing methods.

Text mining shares many characteristics with classical data mining, but also differs in many ways. On the one hand, many classical data mining tasks and algorithms, such as value prediction or decision trees, are irrelevant or ill suited for the textual application. On the other hand, there are special mining tasks, such as concept relationship analysis, which are unique to text mining. In addition, the unstructured form of the raw data necessitates special preprocessing for extracting the main features of the text. Thus, it is necessary to provide special tools and systems geared specifically to text mining.

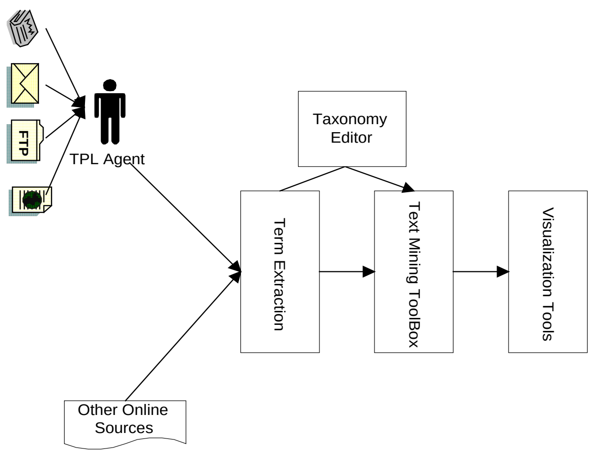
Current text mining systems (e.g., the KDT system [7,8], FACT [6] and WEBSOM [11]) have a rigid architecture. The system provides the user with one or more fixed analysis tools, with no ability to combine between tools, or to construct new analysis paths. Thus, the analysis process is limited to the small predefined analysis tasks envisioned by the creator of the system.

In this paper we describe TextVis, a new integrated visual system for text mining. TextVis takes a multi-strategy approach to text mining, allowing the formation of arbitrarily complex analysis tasks. The system provides the user with a set of *basic-tools* as building blocks, and complex analysis tasks are constructed by combining the basic tools. The process may be interactive, defining and refining the analysis process ad-hoc. The system provides a visual interface for constructing the analysis path. Using the visual interface, basic-tools are represented as drag-and-drop icons (from an icon pallet), and the combination of tools is executed by drawing arrows between icons on the workspace. In addition, TextVis provides an elaborate set of browsers, including IR (Information Retrieval) tools, for viewing the data and the analysis results. The browsers are integrated into the analysis process, and may be used as active filters. Thus, TextVis provides a knowledge discovery framework which integrates advanced mining tools with traditional IR.

TextVis is a complete knowledge-discovery-in-texts suite, supporting the entire discovery process, from the data collection step, through the linguistic analysis of the text, to the mining analysis, and the visual presentation of results.

The TextVis System builds upon the experience that was gained from building the KDT system [7,8], FACT [6], and Explora [9,10].

Outline. The rest of this paper is organized as follows. We start by providing a general overview on the architecture of the system. Next we describe the different components of the system, including analysis tools and visual browsers. Finally, we describe the formation of complex analysis paths, using the visual workspace. The examples we provide were created using a set of 52,000 articles of Reuters Financial



News from 1995-1996.

Fig. 1. TextVis System Architecture

2. System Architecture

TextVis takes a multi-strategy approach to text mining, and enables building complex text analysis schemas based on the basic components provided. The schemas are built by dragging functional icons from a tool-palette onto the workspace and connecting them according to the desired flow of information. TextVis is a complete text mining system supporting the entire mining process, starting with obtaining the documents from various online information sources and ending with a set of visual browsers to view the results of the various data mining tools. The architecture of the TextVis system is show in Figure-1.

The TextVis system is comprised of four main components: an information gathering agent, a document preprocessing module, mining tools, and visualization tools. It further contains a taxonomy editor to aid in creating taxonomies which support the term extraction and mining tools. We now describe each of the four system components.

Information Gathering: The TPL (Text Parsing Language) Agent is used for gathering information. This agent can access various types of information sources including FTP Sites, WWW Sites, email messages, and newsgroups. The agent has a special script language for specifying the exact task to be performed. Each script includes the URL of the information source, a control mechanism for interacting with the specific site and some parsing guidelines for identifying the relevant information. TPL packages the information extracted in a format suitable for further analysis by the TextVis system. We have experimented with accessing the results of a search engine(Alta Vista), a news wire (Reuters) and a special knowledge sources (IBM Patent Server).

Document Preprocessing: After the document collection is identified, the system starts by preprocessing and extracting terms from the documents. Each document is represented by a set of terms which characterize its content. The terms can be accompanied by their frequencies in the document and a weight signifying their importance. The system stores the following attributes with each document: the date, title, author/source, body of text, and a set of extracted terms. The system handles two types of terms: Tagged Terms and untagged Terms. Tagged Terms are terms that are already categorized, like "<Inventor>: David Lewis". Usually tagged terms are fetched from HTML or SGML files where the HTML tag is used as the tag of the term. The TPL agent is able to extract such tagged terms. Untagged terms are terms that were extracted from an untagged portion of the document, usually the document text itself. The output of the preprocessing phase may be saved as a flat file, SGML file, binary file or in a relational database. The data module of TextVis supports data sources in any of these formats and presents a uniform interface to the mining tools. The original source type is invisible to the rest of the system. The data module implicitly builds an inverted file upon loading any source. This is relied upon by different modules of TextVis enabling fast retrieval of all documents containing a set of entities. The module further supports fast retrieval of a given document's term list and original text via its identification number. The free interaction of the data module with the different mining tools emphasizes the modularity unique to TextVis.

Mining Tools: The text mining tools provided by TextVis can be grouped into two classes: Filters and Analyzers. *Filters* are modules which output a subset of the document set. They are typically coupled with a visualization tool for exploring the input set and choosing subsets for further mining. As text corpora are usually very large and diverse, filters are essential for reducing the size of the input and achieving efficient data mining. The output of a filter is handled by TextVis as any other document source, and can be the subject of further analysis. *Analyzers* typically apply a mining algorithm on their input set and extract information regarding that set. The format of this information differs for the different mining tools. Whereas filters are usually used to aid in further mining, analyzers are often results within themselves containing important analytical information.

The Filters currently offered in TextVis are:

1. Query Tools: the output is a set of documents that satisfy a given query.
2. Clustering Tools: the tool gets as input a collection of documents and produces a set of clusters. Each cluster contains documents that are similar to each other using a predefined metric.

The analyzers currently offered in TextVis are:

1. Distribution Based Tools: the system can compute the *profile* of any entity found in the document collection. Typically entities are people, companies, organizations, concepts, countries etc. A profile is the co-occurrence distribution of the given entity with closely related entities. The system can also compare between a set of profiles to find similarities and differences.
2. Frequent-Set Based Tools: the system can generate all frequent sets or only those that satisfy a set of content based constraints. See Section 3 for details.
3. Association Based Tools: the system can generate all associations [1,2] that meet a set of constraints.

Visualization Tools: TextVis offers a set of visual browsers to aid the user in the interactive process of analyzing the data. The system offers browsers for each entity type (e.g., documents, clusters of documents, frequent-sets, association rules, concept distribution trends etc.). Entities may have multiple browsers to allow different visualization capabilities. When browsing a pattern found by the system, the list of the documents, and actual documents, that support the pattern may be viewed.

TextVis is an object-oriented visual system that enables the user to draw visual analysis diagrams. Each diagram starts with one or more document collections. The system is able to read documents from various sources including flat files, SGML files, binary files and files stored in databases that support the ODBC standard. In addition to the document collections, each diagram may include a sequence of filters, analyzers and visualization tools. Each operation produces an intermediate object, which can then be the input for another tool. This enables the spontaneous building of analysis schemas where, upon seeing an interesting pattern, the user may further research that pattern by using it for input to other tools. Intermediate objects may be a collection of frequent sets, a set of association rules, a distribution, or simply an intermediate document collection. Each object has an associated input and output format. The system will allow connecting objects only if the output of one object will match the input of the other object. Trying to combine incompatible objects will result in changing the cursor to a no-entry sign.

3. Text Mining Operations

As was mentioned above, TextVis provides the user with a palette of tools that can be used for building complex text analysis schemas. In this section we highlight a number of the tools, some of them new, provided by the TextVis System.

3.1. Query Module

This module provides the basic functionality of an information retrieval system. The user can enter a query and receive all documents satisfying that query. The resulting set of documents can then be "exported" for further analysis. The export operation creates a new icon in the graphical work area, representing the new sub-collection of documents. This new sub-collection is handled by the system as any other document source. It can therefore be subject to further analysis.

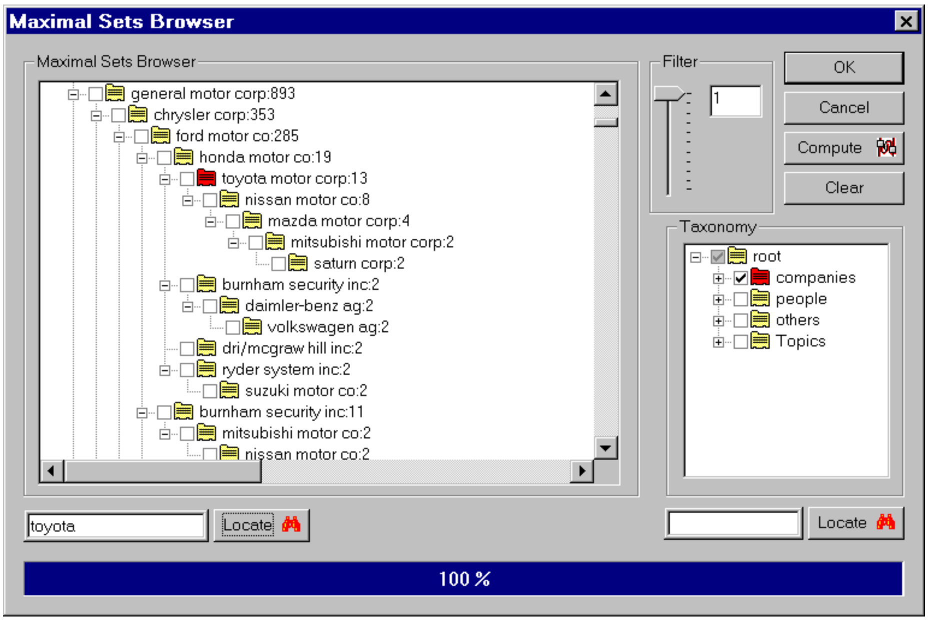


Fig. 2. *Maximal Set* browser

3.2. Clustering

Any set of documents (either an original set, or the result of any of the other tools) can be clustered by the system. The user may choose from one of four classical clustering algorithms: Hierarchical, k-means [12], Buckshot, and Fractionation [3]. Following the clustering, the user can select any number of documents, either individual documents or whole clusters, and export them for further analysis.

3.3. Frequent-Set Based Tools

Frequent Sets [1] are sets of terms that co-occur frequently in the collection (above a user defined threshold). Although originally defined as an intermediary step in finding association rules, we found that frequent-sets contain much interesting information within themselves. *Maximal sets* [5] are sets of items relating to a user specified topic. We say that a document supports a given set if the document contains the terms of the set and there is no larger set which the document supports. The maximal sets browser in TextVis presents the output in a tree format, giving an effect similar to clustering. The root of the tree contains the term which appears in the most sets. The sub-tree of any node is comprised of sets containing the term in the node. The first son of a node is the term which appears most in the sets containing the term's node.

In Figure 2 (previous page) we see an example of the maximal sets associated to the subject "companies". General Motors appeared in 893 documents, of them 353 included Chrysler and so on. The sub-tree of General Motors is a cluster connected to car companies and their co-occurrences. The documents relevant to a cluster are easily viewed by double-clicking on the appropriate node. This tool is also used to aid in generating taxonomies for the system.

4.2. Association-Based Tools

These tools accept frequent sets as input, and output association rules. Association rules are rules of the type $A \Rightarrow B$ where A and B are sets of entities. The system can generate all associations that meet a set of constraints. Constraints are either threshold-based (confidence and support) Some constraints are threshold-based (confidence and support) and relate to the contents of the association.

4. Text Analysis Schema

The uniqueness and power of TextVis is its ability to build elaborate analysis schemas. Each such schema is combined of Data Sources, Analyzers, Filters and Visualizers. The user builds an analysis schema by dragging icons from pallets and connecting them with arrows. Each icon represents one of the TextVis modules and has its own particular settings, depending on the nature of the object. Data Source objects settings define the files from reach to read the data. Algorithmic components settings are used to set the appropriate parameters needed to run the algorithms.

In Figure 3 we see a sample analysis schema. We started by selecting a Data Source and attaching it to the Reuters Financial Collection from the years 1995-96. We used two analysis paths. In the upper path we first applied clustering of the entire collection. Examining the clusters, we identified a cluster consisting of documents regarding "joint ventures". We where interested in analyzing this cluster. However, the cluster was too large for efficient exploration. Thus, we exported it to a Subset Data Source which was then re-clustered. The resulting clusters provided us with a grouping of the articles regarding "joint ventures" by industries. In particular, the system identified a cluster of documents about joint ventures of cable TV companies, and another cluster on oil companies (due to lack of place, the results are not shown). In the bottom path we used an IR filter to select all documents regarding high-tech industries. We obtained a subcollection (marked Data Set in the visual environment). We then applied a clustering algorithm to this subcollection, and exported two sub-collections of documents. One of these collections was used for analysis by data mining tools. Frequent sets were built and association rules generated. Browsers were attached to icons storing the frequent sets and rules at the end to enable viewing the intermediate and final results.

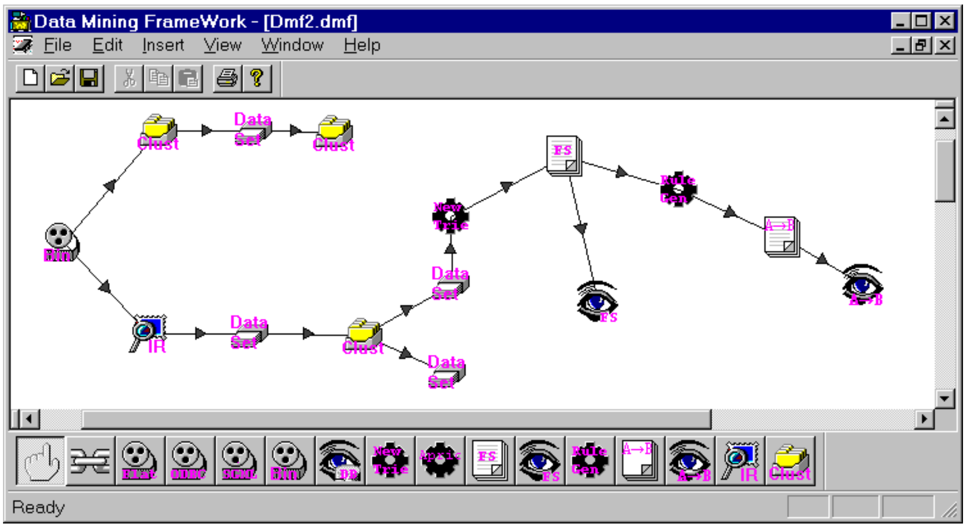


Fig. 3. Analysis Schema

5. Conclusions

TextVis is a second-generation text mining system in that it provides an integrated environment for text mining. The system enables the user great flexibility in combining a variety of different tools. A key feature of the system is the ability to construct complex analysis paths comprising of both classical information retrieval tools for filtering and powerful data mining tools for extracting interesting information. The analysis process is interactive, so that the analysis path may be refined based on the output of the previous steps. At any point in the analysis, several non-exclusive options are available to the user. More than one tool may be used originating from a single point. Another strength of the system is the clear separation between algorithmic tools and browsers. This enables each user to view the results of the algorithmic tools with a browser that suits his/her needs. The major benefit of this approach when compared to previous first-generation text mining systems such as KDT, FACT, and WEBSOM is that the user is no longer limited to the individual tools provided by the system. The ability to combine tools provides an unlimited number of analysis schemas that can be built. Due to the high level of modularity, adding new components to the system is quick and simple, making the system readily expandable. New modules are added with ease and interact with the overall system according to defined uniform interfaces. The system is equipped with a TPL agent which enables a constant feed of articles and documents. The tight integration between the agent and the text mining modules provides a very powerful tool for mining the vast information available on the Internet.

The system allows storing schemas in a library. The schemas can then be reused by other users. An organization can thus create a library of common analysis tasks. Schemas can be customized to fit the exact needs of each user, without the need to rebuild the full analysis path each time.

In the future, we plan to add more components to the system, including information extraction tools and support for a larger variety of data sources

References

1. Agrawal A., Srikant R.: Fast algorithms for mining association rules. In: Proceedings of the VLDB Conference, (1994).
2. Agrawal A., Imielinski T., Swami A.: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD Conference on Management of Data, (1993) 207-216.
3. Cutting D. R., Karger D. R., Pederson J. O., Tukey J. W.: Scatter/Gather: a cluster-based approach to browsing large document collections. In: Proceedings of the 15th International ACM SIGIR Conference on Research and Development in Information Retrieval, (1992) 318-329.
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., Aumann A., Amir A., Zilberstein A., Kloesgen W.: Maximal Association Rules: a New Tool for Mining for Keyword Co-occurrence in Document Collections. In Proceedings of the 3rd International Conference on Knowledge Discovery (KDD),(1997) 167-170.
6. Feldman R., and Hirsh H. "Exploiting Background Information in Knowledge Discovery from Text, ", Journal of Intelligent Information Systems, (1997).
7. Feldman R., Dagan I., Kloesgen W.: Efficient Algorithms for Mining and Manipulating Associations in Texts. In: Proceedings of EMCSR96, (1996).
8. Feldman R., Dagan I.: KDT - knowledge discovery in texts. In: Proceedings of the First International Conference on Knowledge Discovery (KDD), (1995).
9. Klösgen W.: Efficient Discovery of Interesting Statements. The Journal of Intelligent Information Systems, 4(1) (1995).
10. Klösgen W.: Explora: A Multipattern and Multistrategy Discovery Assistant. In: U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, (Eds.) Advances in Knowledge Discovery and Data Mining, MIT Press, Cambridge, MA (1996).

11. Lagus, K., Honkela, T., Kaski, S., Kohonen, T.: Self-organizing maps of document collections: A new approach to interactive exploration. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD), (1996) 238-243.
12. Rocchio, J. J.: Document retrieval systems – optimization and evaluation. Ph.D. Thesis, Harvard University, (1966).