Visualizing The Semantic Content of Large Text Databases Using Text Maps

N93-25982

Nathan Combs
TASC
Reading, MA 01867
ncombs@tasc.com or (617) 942–2000

Abstract

A methodology for generating text map representations of the semantic content of text databases is presented. Text maps provide a graphical metaphor for conceptualizing and visualizing the contents and data interrelationships of large text databases. Described are a set of experiments conducted against the TIPSTER corpora of Wall Street Journal articles. These experiments provide an introduction to current work in the representation and visualization of documents by way of their semantic content.

Introduction

This paper presents a methodology for deriving text-map representations of large text databases. Text maps are useful graphical metaphors that can aid users in visualizing the semantic contents of large text databases. The text map graphical metaphor relates to Artificial Intelligence (AI) research in two substantive ways: first, text maps are often generated using "neurally inspired" computational paradigms; second, they can be effective tools for relating at a high level with the complex knowledge structures generated by AI systems. It is this capacity as a vehicle for communicating intuitions about large quantities of highly interconnected knowledge that is of primary interest. This kind of capability is seen as relevant to NASA activities in the areas of text processing (Driscoll et al. 1992), information retrieval (Rorvig 1991), and knowledge understanding. Specific applications would include searching and navigating amongst the contents of regulatory document databases as well as technical and scientific document collections. Additionally, text-based retrieval methods can be generalized to other information domains. Thus, for example, databases that link textual information to geographic and image objects can make use of textual features to organize and access these objects (e.g., Carlotto 1992). Using textual information as a key data source can be useful for describing a range of data for two reasons: first, textual information tends to be abstract — which facilitates high level data classification; and second, textual descriptions are expressive — and are thus robust across applications and data types.

Also introduced in this paper are preliminary results of a set of text-map experiments conducted against the TIPSTER corpora of Wall Street Journal articles. These experiments made use of a simple statistical text pre-processor. More relevant to this paper, however, is the follow-on discussion describing how the simple experimental system is currently being extended with a semantic-based text preprocessor.

The discussion in this paper is framed conceptually in terms of two levels of understanding: a micro-scale level of understanding which is concerned with finding the "gist" of the semantic content of individual documents; and a macro-scale level of

understanding which is concerned with integrating the mosaic of individual document interpretations into a larger meaning. Abstracting from the micro scale to the macro scale is the function of such tools such as inference (rule) generators, database knowledge "mining" techniques, and visualization maps. It is the latter technique that is proposed by this paper.

The proposed visualization approach is of general relevance to the NASA community working with "vector-product" information retrieval systems (Rorvig 1991). Furthermore, this work is of specific relevance to NASA projects in text browsing and retrieval such as Kennedy Space Center's QA system (Driscoll et al., 1992) and other work conducted in the Astrophysics Data Facility at Goddard. As a knowledge abstraction technique, text maps generate simple topographic knowledge structures for interpreting the contents of a database. From these structures, users infer associations and similarities amongst database items.

Visualization

There are two challenges confronting text processing systems working with large document databases: how to extract meaning from documents; and how to integrate and represent this extracted meaning. Browsing large text databases whose contents are not fully characterized, or whose contents are subject to change, requires information abstraction tools. Advocated here is an approach that integrates information about the meaning of multiple documents into a single gestalt which can be graphically and intuitively conveyed to a user. The contrast between this approach to information abstraction and other artificial intelligence methods is made in a later section.

With text database understanding, there are two separate but related concerns:

1.) How can the aggregated output of a text preprocessor be concisely presented?

2.) What kind of higher abstraction can be used to express the interrelationships of documents?

One method for representing document classifications is by the RANKED LIST. With the ranked list the documents in the database are linearly ordered according to how well they match a target set of concepts. The more relevant a document is to a target set of concepts, the higher in the list it is positioned. How a document relates with other documents with regard to a specific set of criteria can be communicated by this list. One example of a ranked-list system that allowed users to select database objects whose descriptions best fit a user query is given by Rorvig (1991). With this system, if the retrieved objects do not match the query, a user can extend the search and look at objects which are "like" the best matching objects found in the list (relevance feedback).

One drawback of the ranked list representation is that the concepts that are being searched for need to be known before the search is implemented. In circumstances where the significant concepts or terminology are not well understood, a ranked linear representation can be restrictive: it does not easily communicate how documents differ from a query, and to what extent. Thus, with Rorvig's example, the results of the relevance feedback are not integrated into a single representation which communicates the relationship of the queries with the contents of the database. Instead, what is presented is a series of disparate "snapshots" of the database as it is evaluated against an evolving query.

Documents can also be represented hierarchically using dendrograms, or trees (a product of single-link clustering, for example). Each leaf in the tree denotes a document. The tree depicts how the documents are incrementally aggregated into ever larger groups: links connect documents or groups of documents to their nearest neighbors. Hierarchical structures are interpreted visually by sequentially traversing their component links: relationships are identified by paths through the cluster hierarchy.

An alternative display for hierarchical document structures that de-emphasizes their sequential interpretation has been proposed by Schneiderman (1991) in his work with tree-maps. With tree-maps, database objects are denoted by surface-filled, color-coded rectangles. Rectangles are colored to show the object type, and the rectangle areas indicate how relevant that object is to its type. While tree-maps are easier to grasp visually than a sprawling tree structure, they, like dendrograms, do not easily communicate how arbitrary documents are related.

The text map is proposed here as an alternative to both the hierarchic structure and the ranked list. This representation provides a comprehensive picture of all documents in a database, unlike the ranked list, and is preferred to hierarchical cluster representations because of its intuitive use of the two dimensional viewing surface.

While hierarchic structures can serve a useful role in facilitating database search and retrieval and thus may underlie the data organization of any representation (van Rijsbergen 1976), for many text database comprehension tasks, hierarchical document displays may be counter-intuitive:

1.) hierarchies require sequential interpretation.

2.) hierarchies restrict comparisons between documents.

3.) re-balancing hierarchic structures with new documents can cause dramatic changes to the structure.

A more basic complaint, however, relates to the hierarchic classification methodology itself: because the implicit goal of this method is to partition data into disjoint sets, it cannot easily represent structures derived from statistical distributions (Kohonen 1982). When viewing document distributions, how the parts relate to the distribution carries meaning. It is this connectivity between the documents that is undermined by hierarchic representations: rather than emphasizing how the parts are connected to the whole, what is emphasized is how the whole is decomposed into ever shrinking sets.

In contrast, the text map approach assumes that the interrelationships between documents are significant and representable. It is meant to provide a macro perspective of the database that is intuitive as well as abstract. Because it avoids representing the semantics of the extracted text, it is a visualization method that can be used across a variety of applications and databases.

Examples of vector-based systems that use visualization maps to represent database information analogous to the approach described here include Carlotto (1992) and Chang (1990).

Text maps

Text maps are forwarded here as a visual metaphor for graphically communicating the taxonomy of the contents of large text databases. Using text-maps, documents are classified contextually with associations between documents being implied by proximity. The text visualization (TEXTVIZ) approach assumes a vector representation of the meaning of documents: each vector encodes a set of features which characterize the content of each document. Vector components index individual document features and vector component values denote the pertinence of a feature to a particular document. As will be illustrated later, the actual semantic content of a feature is determined by the preprocessor. Preprocessors that use a semantic model of a domain can generate semantically meaningful features for that domain. Whereas preprocessors that use non-semantic models, e.g. statistical systems, the features correlate with other properties of the document such as word distribution.

Thus, for each document in the database, there exists a corresponding vector description of its content. These vectors, points in vector space, are then projected onto a two-dimensional text-map surface for display.

The TEXTVIZ procedure converts the information extracted about the content of each document into a numeric vector, a signature. Differences in meaning between documents is reflected by differences in their vector patterns. Thus, from a macro perspective, the organization of the database is inferred from observing how the signature patterns vary across all documents.

Ultimately, documents are presented to the user as points on a viewing surface or map; distances between points represent the difference in the estimated meaning of the documents. It is the relative similarity (dissimilarity) of the meaning of documents that is graphically represented by the text map. Thus, there are two levels of abstraction contained by the TEXTVIZ approach: first, meaning is abstracted from text using a text preprocessor; second, descriptions of all documents are aggregated, abstracted, and then displayed graphically. The abstraction process is accomplished by projecting the document signature vector into a two dimensional visual space (x and y coordinates on the text map).

Figure 1 provides a system overview of the text visualization process. Figure 2 is an example of a text map that was generated by a surrogate-coding system developed at TASC (Carlotto 1992). In this map, individual documents are represented by word labels.

Visualization and Artificial Intelligence.

The TEXTVIZ approach intersects disciplines in AI in two ways: first, neural network procedures, including the Kohonen self-organizing map (1984,1982) can be used to generate visualization (text) maps; second, visualization maps can serve as a powerful means for integrating the output of AI semantic reasoning processes. The semantic text processor that is introduced later in this paper serves as one example. Similarly, as a knowledge abstraction tool, the function of visualization maps parallels many database rule induction, data classification, and data clustering methods (Piatetsky-Shapiro and Frawley 1991).

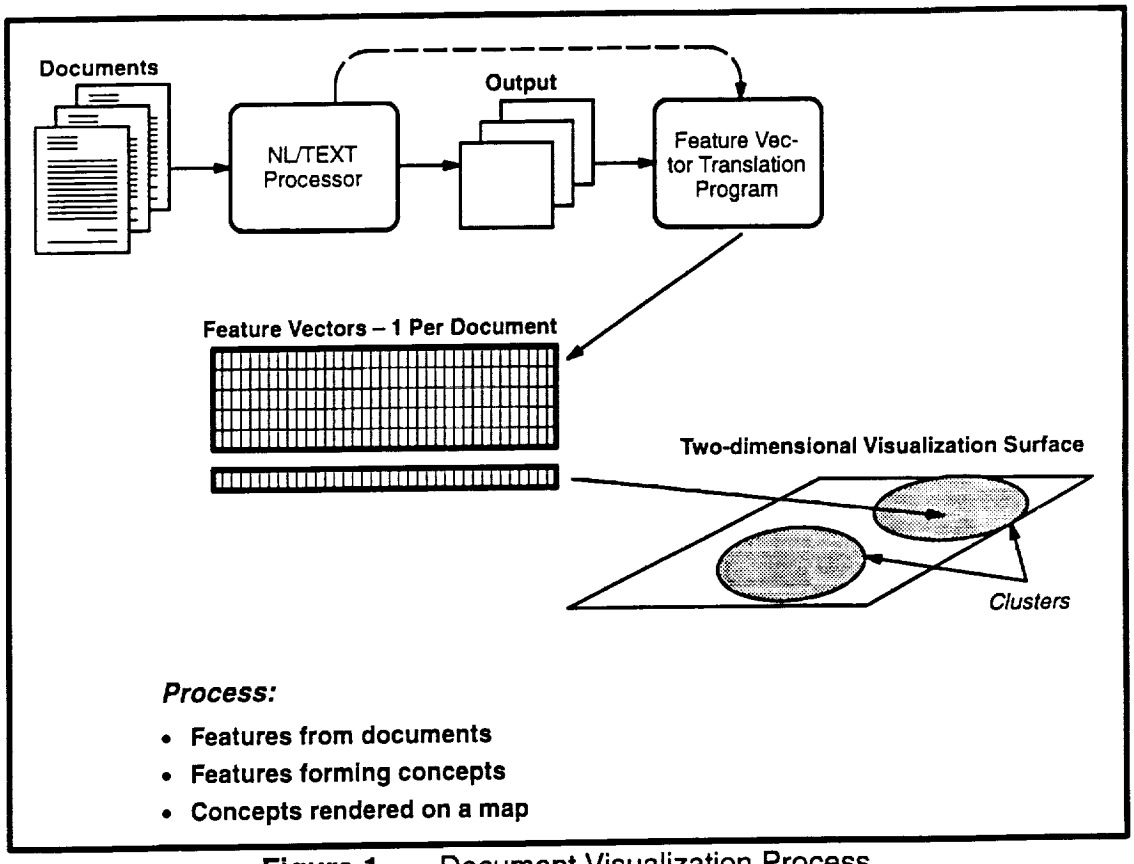

Figure 1. Document Visualization Process

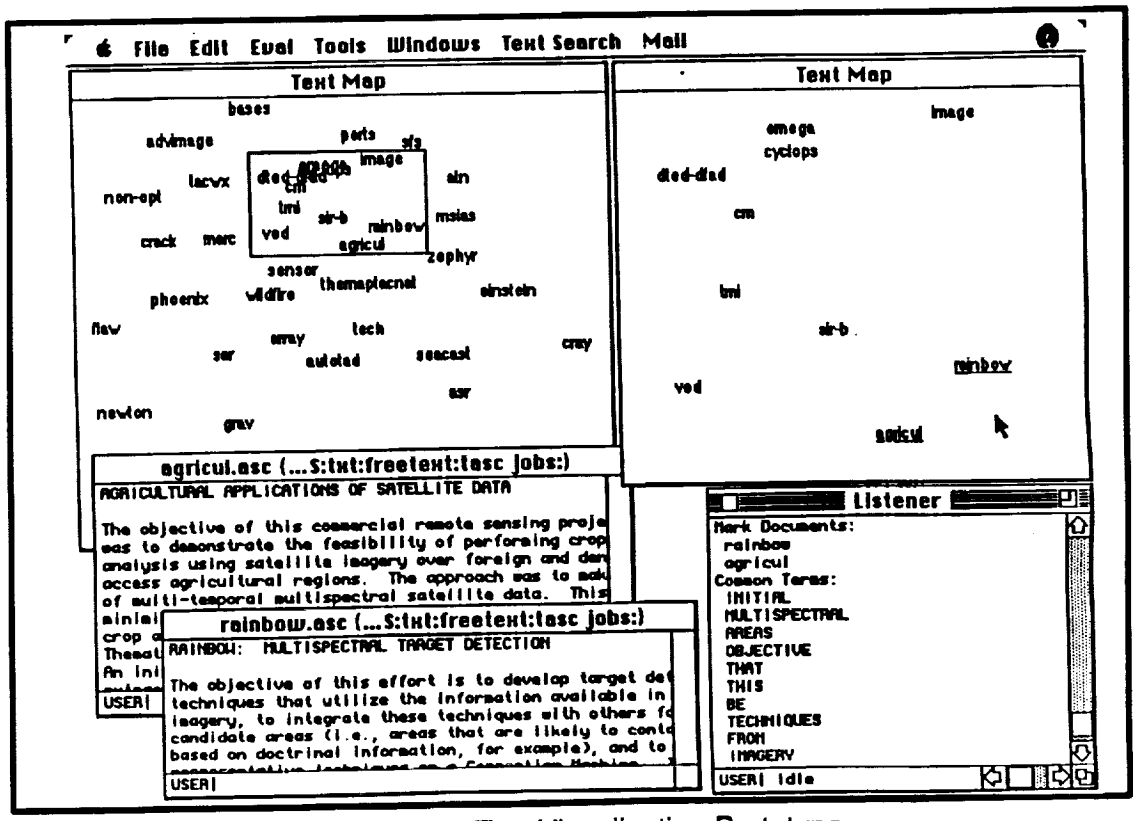

Figure 2. Text Visualization Prototype

Text visualization, however, does differ from many other AI-based database knowledge discovery methods by its abstract representation. Text maps provide a high-level topological portrait of the database that uses spatial distance on the display surface to measure the degree of association between database text objects. This approach is advantageous with large databases of objects where measurements of similarity are reducible to a scalar measure of "distance." When more complex data relationships exist and their description is sought, a visualization map would be less effective because of its inability to express fine-grain dependencies between individual items.

For large text database navigation purposes, a representation that trades fine-grain document contrasts for a cohesive global picture of the database is desirable. There are two reasons for this. First, real-world text processing often requires operation over broad subject domains using unconstrained natural language text. This means that the information extracted from documents may be too coarse to infer exact relationships. Second, the information navigation problem is primarily one of localizing areas of search through a process of iterative user guidance. This process must initially be coarse-grained because most users who browse a database either do not have a concise description of what is being searched for, or an exact understanding of the contents of the database. The need for broad-scoped database navigation is emphasized by databases whose contents are volatile and subject to change.

Early Experiments

In order to speak tangibly of the TEXTVIZ approach, an initial set of experiments using text-maps will be described here. This discussion will introduce the text-map visualization method. Then, in the next section, this discussion will be broadened to include current work. The following initial experiments were conducted using a subset of the Wall Street Journal documents contained by the 1992 NIST TIPSTER corpora (2 gigabytes) of documents.

In these experiments, the information content of documents was crudely estimated using a statistical procedure based on word-frequency counts (Figure 3). The meaning of a document was estimated by examining the frequency profile of significant words that occurred within that document. For these experiments, the set of significant words coincided with the set of words that occurred infrequently in a training corpus of documents. The frequency threshold cut-off varied across experiments and was arbitrarily selected. Low frequency words were used because of an assumption that they were generally more indicative of the meaning of a document than high frequency words. Early work in text processing (e.g., discussion in van Rijsbergen 1979) bears this out.

Once the set of significant words was selected, these then become the set of word features by which the content of documents in the test corpus would be characterized. This approach at estimating the content of a document is analogous to the vector-based score-and-rank systems used by Salton (1971) and Stanfill and Kahle (1986).

Upon completion of analysis of the training corpus of documents, a set of feature words were identified which were then used to analyze a test document set. The results were portrayed on a text map. For most experiments, the training and test document sets were identical. All the documents were analyzed, and for each document, a feature vector

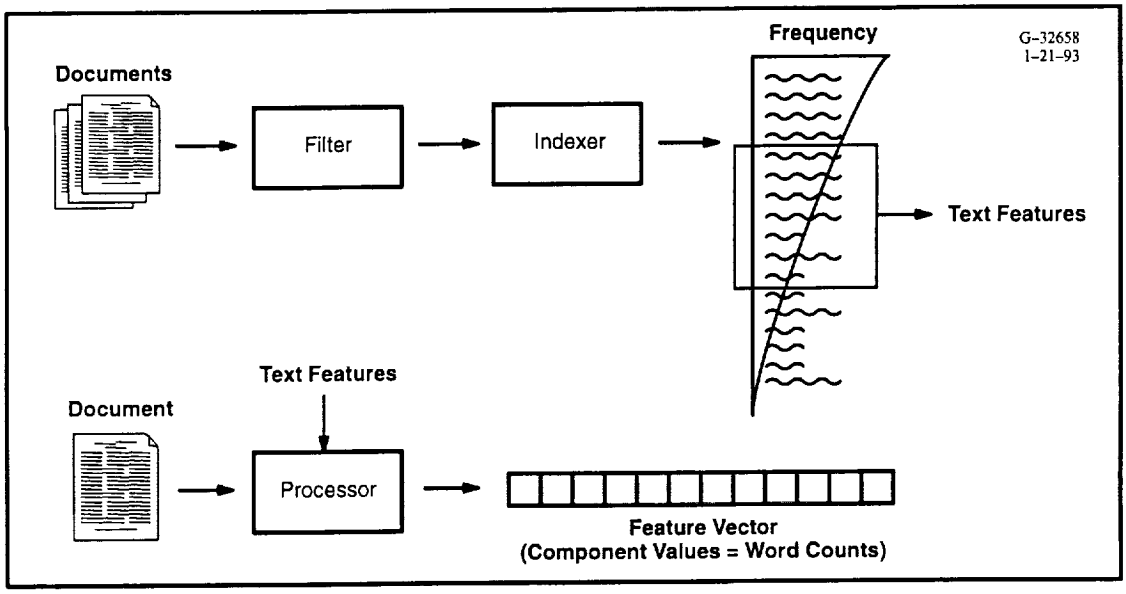

Figure 3. Text Processing: Using Word Frequencies To Estimate Meaning

was constructed. Every feature word indexed a specific component in the feature vector. Component magnitudes encoded the actual number of times a word occurred within a given document.

The procedure for analyzing the information content of a document worked as follows:

1.) An initial vector of length N (= size of feature set) was created for each document. All component values in the vector were set to zero.

2.) Every word in the document was examined; if it were a member of the feature word set, the indexed vector component value was incremented.

3.) Every vector for each document was normalized with respect to the total word count for that document.

After completion of the document analysis phase, the feature vectors were then projected onto a two dimensional visualization map. Documents were positioned on the map to reflect how close in vector space their feature vectors were. Documents of similar feature word profiles, and hence with similar estimated content, were placed close together while documents of dissimilar content were placed farther apart. Feature vectors were projected from vector space onto a two-dimensional visualization surface.

Initially, a self-organizing map (SOM) was used to implement the vector space to map projection. A SOM is an algorithm that simulates a planar "neural" network of interconnected processing units (Kohonen 1982, 1984). These processing units converge to an "accurate" portrait of the database through an adaptive process based on a competitive neural network learning procedure. However, given the large number of features that a feature-word document analysis procedure can generate (some experiments had as many as 12K feature words), a SOM was too costly to use. The size of the connectivity matrix and the number of processors required for high resolution maps restricted its use.

For this reason, an alternative, stochastic steepest-descent approach was used. While it was similar to Sammon's non-linear mapping algorithm (1969), it also made use of a stochastic search regime analogous to a simulated annealing procedure (Kirkpatrick 1987). An overview of the algorithm is as follows:

1.) Choose an initial random map configuration.

2.) Calculate the error-per-map configuration by measuring the deviation of the relative point distances on the display map from the relative distances of the vectors in vector space.

3.) Effect random perturbations to the map configuration; seek to minimize the error iteratively by using gradient descent. To minimize the possibility of local minima traps, a global "annealing" regime was in place that allowed the system to sometimes accept non-optimal changes (a monotonically decreasing probability of acceptance).

The product of these experiments was a set of text-maps such as the one given by Figure 4. By visually analyzing these text-maps and relating their topographical distribution with their interpreted differences in content, it was clear that there were several deficiencies with this approach.

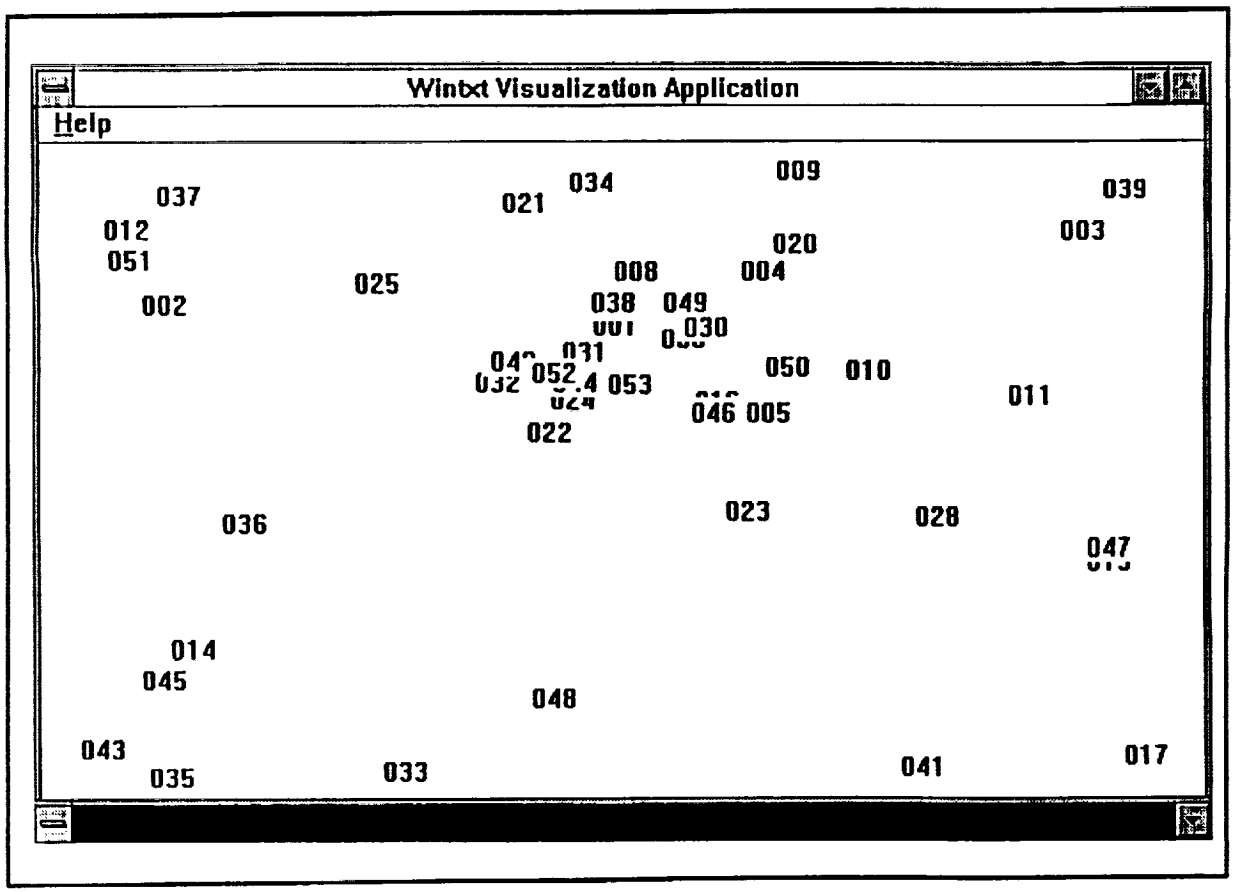

Figure 4. Visualizing WSJ Articles Using Word Frequencies

First, it was apparent that feature word profiles were at best weak predictors of the content of documents. This would seem to be especially true for large text sets. Other experiments suggested that larger linguistic units such as phrases would be better predictors of document content.

Second, it was difficult to predict which frequency ranges were best for selecting feature words from an arbitrary training corpus. Within training corpora whose documents were narrowly focused upon a few distinct subjects, idealized feature words tended to occur more frequently than they would in wide-ranging document corpora. Similarly, technical documents also tended to exaggerate the frequencies of idealized feature words.

Thus, document corpora can bias word frequency distributions in ways that are unpredictable in advance. The problem was one of predicting distribution biases without knowing anything about the actual semantic content of the documents.

And finally, this approach to generating text features did not scale-up well to larger, more discriminating applications. In other words, expansion of the feature word set to include more content words did not necessarily make the system more robust (better able to recognize a subject), and less brittle (cover more subjects). Feature word sets can be expanded by accepting more words from the training corpus (broadening the frequency range) or through dictionary or synonym expansions. The problem with word expansions, however, is that while it does improve the system's ability to recognize content words, it also introduces substantially more "noise words" into the system. The introduction of noise words distracts from meaningful comparisons between documents. Strategies for minimizing the number of "noise words" such as stemming (to remove redundant inflected forms of words), and filtering of low content words such as determiners and prepositions do exist: while they do help, they do not solve the problem.

Visualization by Semantic Content

As was suggested by the earlier experiments, more predictive text features would be needed if documents were to be accurately characterized. The current section will describe an approach that seeks to estimate the meaning of documents from the semantic content of recognized words and phrases found in a document. The described approach is undergoing implementation at TASC and will be tested in 1993 against the TIPSTER corpus of documents. This system will introduce key concepts about how a varied distribution of semantic content information from a document can be hierarchically integrated into a single feature vector description which in turn is represented on a text map.

The algorithm described here is similar to S. Gallant's work (1991, 1992) with several major modifications. It is hypothesized that these modifications will improve document discriminations by providing mechanisms to identify larger linguistic units, to better localize meaning within documents, and to provide a more accurate representation of multiple subjects within documents. These mechanisms are designed to operate in concert, to enhance intra-document content discrimination and representation. This is important when trying to represent the content of a large document without "blurring" together the subjects contained within that document.

Words, in the described model, are defined by vectors of semantic features. Each feature indexes a vector component and the component magnitude encodes the correlation of that particular word to a semantic class. Syntactic features are also included. Representing the content of a document in terms of a vector of features is analogous to vector-based linguistic models of word-sense and semantic discrimination (e.g., Miikkulainen and Dyer 1991, McClelland and Kawamoto 1986).

Thus, for example, BELGIUM might be defined by the following features (and respective correlations):

```
BELGIUM ->
        +Nation(10)
        +European-Economic-Community(3)
        +North-Atlantic-Treaty-Organization(3)
        +Policy:Agricultural-subsidies(4)
        +Lang:french(5)
        +Lang:flemish(5)
        +Brussels(8)
        +Geo:SAmerica(0)
        +Geo:Europe(8)
        +Geo:NAmerica(0)
        +Geo:Asia(0)
        +Noun(10)
```

Note that while this example illustrates a single definition of BELGIUM, multiple definitions per word may exist. The feature set, as well as specific correlation values for individual definitions, are hand-generated.

Figure 5 provides a simple overview of the system. Level (1) processing involves extracting from the text stream literal string phrases (mostly proper nouns) such as "Wall Street Journal" and "Singapore Airlines." Words from the text stream that do not match phrase rules are then stemmed (suffixes removed). Phrases and stemmed words, or "tokens," are sent on for processing at level (2). At this level, tokens are looked up in a dictionary; each dictionary entry consists of a token and a set of corresponding feature vector definitions (multiple definitions are possible). These vectors can be weighted.

If a text stream token matches a dictionary entry, all defining feature vectors associated with the matching dictionary entry are then aggregated into an ordered list of vectors, or vector stream. Because a word can have multiple definitions, a method of consolidating meaning within document regions is required. The approach adopted at level (3) is to partition the vector stream into bins which correspond to "chunks" of text from a document. The text regions can be of arbitrary size and will generally be delimited by text breaks such as paragraph boundaries. Thus, the size of the vector stream bins will co-vary with the size of the text regions. The size of text regions will be user/application defined, and will depend upon the granularity of the meaning that is sought from the document.

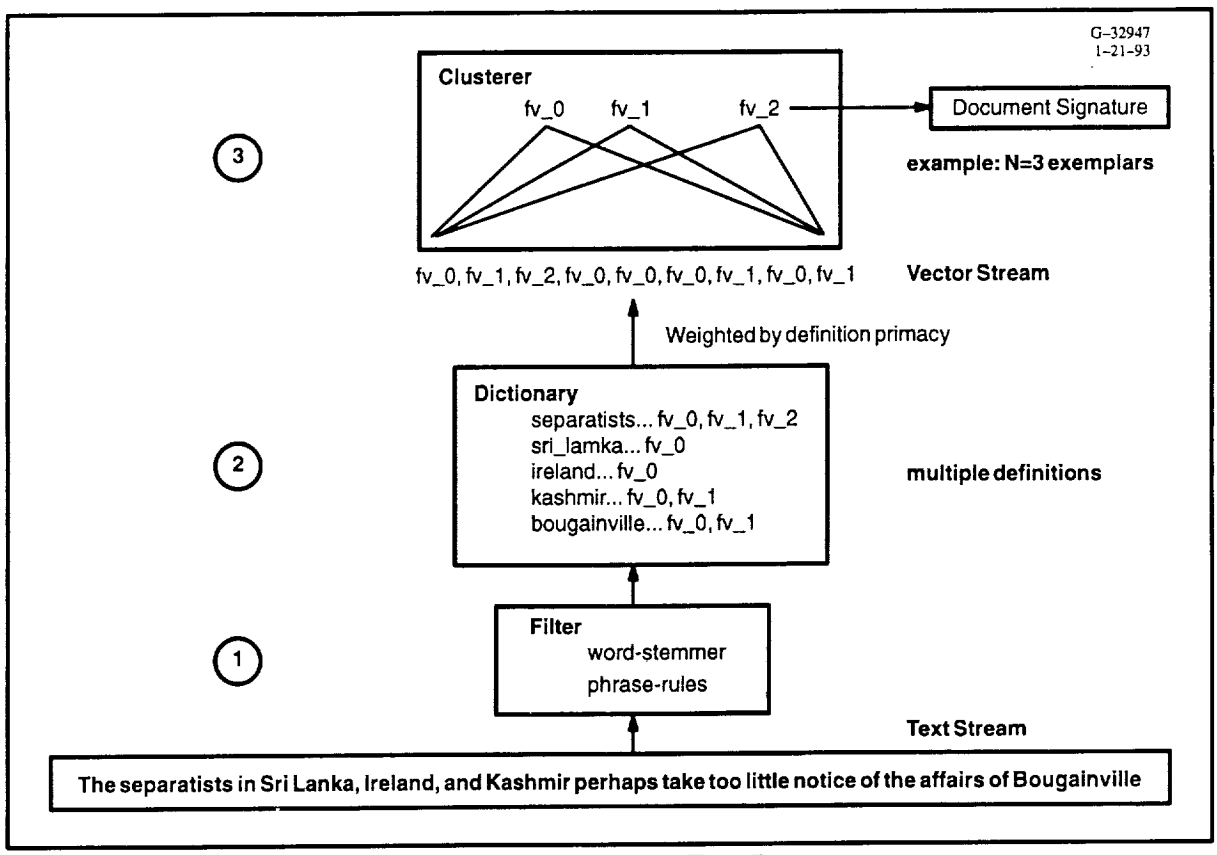

Figure 5. Semantic Text Processor

Within the context of a vector stream bin, a consensus of meaning is sought. It is through a process of consensus building that alternative, inconsistent definitions are pruned. The objective is then to select exemplar definitions (feature vectors) to signify the content of each vector bin. This exemplar will serve as the consensus content vector representing a text region.

The exemplar selection process, carried out for every text region in a document, is an abstraction process that uses context to prune alternative definitions as well as limit the background noise generated by a large population of definitions. The number of definitions can be magnified when working with large text bins and/or working with large dictionaries. In this way, abstraction is used to preserve the integrity of the extracted content information by restricting noise as well as reducing ambiguity.

The exemplar selection process uses a stochastic gradient descent clustering technique that is similar to the algorithm used to generate the text maps (described earlier). Definitions may be weighted to reflect a priori assumptions about which definitions are most significant (e.g., primary definitions are more important than secondary and tertiary definitions). Features within definition vectors may also be used to modulate the exemplar selection process. For example, if syntactic features were included in token definitions, and if it were assumed that certain syntactic categories were more predictive of the meaning of a document than others, then selected syntactic category information (e.g., "+Noun") could be weighted to bias the selection of candidate exemplars.

As stated, the effect of this clustering procedure is to discard extraneous interpretations of tokens as well as to reinforce a consensus in meaning. This consensus is codified by the selection of a set of exemplars used to represent the document. The manner in which a consensus of meaning is reinforced is analogous to Gallant's (1991) ideas on using the local context around a word to disambiguate individual words. In contrast, the approach described here uses a "regional" context, consisting of all words within a text region, to modulate choice of exemplars. A regional approach is generally faster to compute and also serves as a more general mechanism of information abstraction.

The product of level (3) processing is a set of exemplar vectors, one per text region, that are used to symbolize the content of the document. Additional vector abstraction may also be used to post-process the level (3) output in order to consolidate meaning across text regions, i.e. eliminate redundancies in the exemplar list. The approach described here advocates preserving distinct representations for unique regions of text. Thus, the described approach differs conceptually from Gallant's (1992) strategy of vector addition. With this approach, the complete set of exemplar vectors is used to denote the contents of a document. This prevents a document signature from being "blurred" by background feature "noise." Similarly, it provides a means for localizing intra-document content searches and is hypothesized to be of significant value when working with long documents.

Conclusions

As illustrated, text maps can abstractly render general semantic relationships among the contents of large text databases. While this assumes the existence of a semantic model with which to analyze the database text objects, it would not explicitly require such a model for visualization. By separating the semantic interpretation process from the visualization process, users can conceptualize and navigate large complex databases at a high level. Text maps thus provide a graphical and spatial metaphor for reasoning about the contents of large text databases.

Described in this paper is an approach for hierarchically integrating the semantic content of a spatial text stream. Aside from specific relevance to NASA interests in text and document retrieval (Driscoll 1992), such an approach may have broader implications for information management and database knowledge extraction. For example, semantic comparisons between text and other spatially structured data types, such as imagery, could be pertinent.

Furthermore, TASC is investigating how text information can be generalized to other information domains. For example, textually-derived information has been used to augment information from geographic as well as image sources (Carlotto 1992). This work underscores the versatility of the visualization map graphic metaphor, as well as suggests a conceptual interface design for integrating information of diverse types.

References

Carlotto, M (1992, March). "A Text-Based Geographic Information System." TASC white paper.

Chang, S (1990). "Visual Reasoning for Information Retrieval from Very Large Databases." Journal of Visual Languages and Computing, 1. pp. 41–58.

Driscoll, J., J. Lautenschlager, M. Zhao (1992). "The QA System." Preprint of the Proceedings of the Text Retrieval Conference (TREC). Rockville, MD. November 4–6, 1992.

Gallant, S. I. (1992). "HNC's MatchPlus System." Preprint of the Proceedings of the Text Retrieval Conference (TREC). Rockville, MD. November 4–6, 1992.

Gallant, S. I. (1991). "A Practical Approach for Representing Context And for Performing Word Sense Disambiguation Using Neural Networks. Neural Computation 3(3). 293–309.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P (1983). Science, 220. pp. 671–680.

Kohonen T (1984). Self-organization and associative memory. Berlin: Springer-Verlag.

Kohonen T (1982, Oct 19–22). "Clustering, Taxonomy, and Topological Maps of Patterns." Proceedings of the 6th International Conference on Pattern Recognition.

Piatetsky-Shapiro, G., and W.J. Frawley (1991). Knowledge Discovery in Databases. Menlo Park: AAAI Press/ MIT Press.

McClelland, J.L., and Kawamoto, A.H (1986). "Mechanisms of sentence processing: Assigning roles to constituents." In J.L. McClelland, and D.E. Rumelhart (Eds). Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations. Cambridge, MA: MIT Press.

Miikkulainen, R., and M. Dyer (1991). "Natural Language Processing with Modular PDP Networks and Distributed Lexicon." Cognitive Science 15. pp. 343–401.

Rijsbergen, C.J (1979). Information Retrieval. Butterworths: Boston.

Rorvig, M. E (1991, Dec 3–5). "A Vector-Product Information Retrieval System Adapted to Heterogeneous, Distributed, Computing Environments." Proceedings of Technology 2001, NASA conference.

Salton (1971). The SMART Retrieval System — Experiment in Automatic Document Processing. Prentice-Hall: Englewood Cliffs, NJ.

Sammon, J (1969, May). "A nonlinear mapping algorithm for data structure analysis" IEEE Transactions on Computers, C–18(5).

Schneiderman, B (1991, August). "Visual User Interfaces for Information Exploration." Department of Computer Sciences Technical Report No. 2748.

Stanfill, C., and Kahle, B (1986, December). "Parallel Free-Text Search on the Connection Machine System." Communications of the ACM, Vol 29,12.