TopCat: Data Mining for Topic Identification in a Text Corpus

Chris Clifton, *Senior Member*, *IEEE*, Robert Cooley, *Member*, *IEEE*, and Jason Rennie

Abstract—TopCat (Topic Categories) is a technique for identifying topics that recur in articles in a text corpus. Natural language processing techniques are used to identify key entities in individual articles, allowing us to represent an article as a set of items. This allows us to view the problem in a database/data mining context: Identifying related groups of items. This paper presents a novel method for identifying related items based on traditional data mining techniques. Frequent itemsets are generated from the groups of items, followed by clusters formed with a hypergraph partitioning scheme. We present an evaluation against a manually categorized ground truth news corpus; it shows this technique is effective in identifying topics in collections of news articles.

Index Terms—Topic detection, data mining, clustering.

✦

1 INTRODUCTION

D ATA mining has emerged to address problems of understanding ever-growing volumes of information for structured data, finding patterns within the data that are used to develop useful knowledge. Online textual data is also growing rapidly, creating needs for automated analysis. There has been some work in this area [1], [2], [3], focusing on tasks such as:

- association rules among items in text [4],
- rules from semistructured documents [5], and
- understanding use of language [6], [7].

In this paper, the desired knowledge is major topics in a collection; data mining is used to discover patterns that disclose those topics.

The basic problem is as follows: Given a collection of documents, what topics are frequently discussed in the collection? The goal is to assist human understanding, so a good solution must identify topics in a way that makes sense to a person. We also want to enable further exploration, requiring the ability to link topics to source texts. This is related to document clustering [8], but the requirement for a topic *identifier* is closer to rule discovery mechanisms.

We apply data mining technology to this problem by treating a document as a collection of entities, allowing us to map this into a *market basket* problem. We use natural language technology to extract named entities from a document. We then look for *frequent itemsets*: groups of named entities that commonly occurred together. Next, we cluster the groups of named entities, capturing closely related entities that may not actually occur in the same document.

The result is a refined set of clusters. Each cluster is represented as a set of named entities and corresponds to an ongoing topic in the corpus. An example topic is:

ORGANIZATION	Justice Department
PERSON	Janet Reno
ORGANIZATION	Microsoft

This is recognizable as the US antitrust case against Microsoft. Although not as readable or informative as a narrative description of the topic, it is compact and humanly understandable. It also meets our requirement to link to source texts as the topic can be used as a query to find documents containing some or all of the extracted named entities (see Section 3.4).

Much of this is based on existing commercial or research technology: natural language processing for named entity extraction, association rule data mining, clustering of association rules, and information retrieval techniques. The novelty of TopCat lies in how these disparate technologies are combined, plus a few specific developments that have wider application:

- the frequent-itemset filtering criteria (Section 3.2.1),
- the hypergraph-based clustering mechanism, a generalization of the mechanism proposed in [9] (Section 3.3),
- use of information retrieval measures for clustering of associations (Section 3.5).

Although we only discuss identifying topics in text, these developments apply to any problem that can be cast as a market basket.

We next give some background on where this problem originated. In Section 3, we describe the TopCat process from start to finish. Section 4 describes an evaluation of TopCat on the Topic Detection and Tracking project [10] corpus of news articles, including an analysis of how TopCat performs compared to a manually defined ground truth list of topics. In Section 3.1.2, we discuss augmenting the named entities with user-specified concepts. Section 5 concludes with a discussion of ongoing application of TopCat and of future work.

- *C. Clifton is with the Department of Computer Sciences, Purdue University, 250 N. University St, West Lafayette, IN 47907. E-mail: clifton@cs.purdue.edu.*
- *R. Cooley is with KXEN Inc., 650 Townsend Street, Suite 300, San Francisco, CA 94103. E-mail: rob.cooley@kxen.com.*
- *J. Rennie is with the Massachusetts Institute of Technology Artificial Intelligence Laboratory, Cambridge, MA 02139. E-mail: jrennie@ai.mit.edu.*
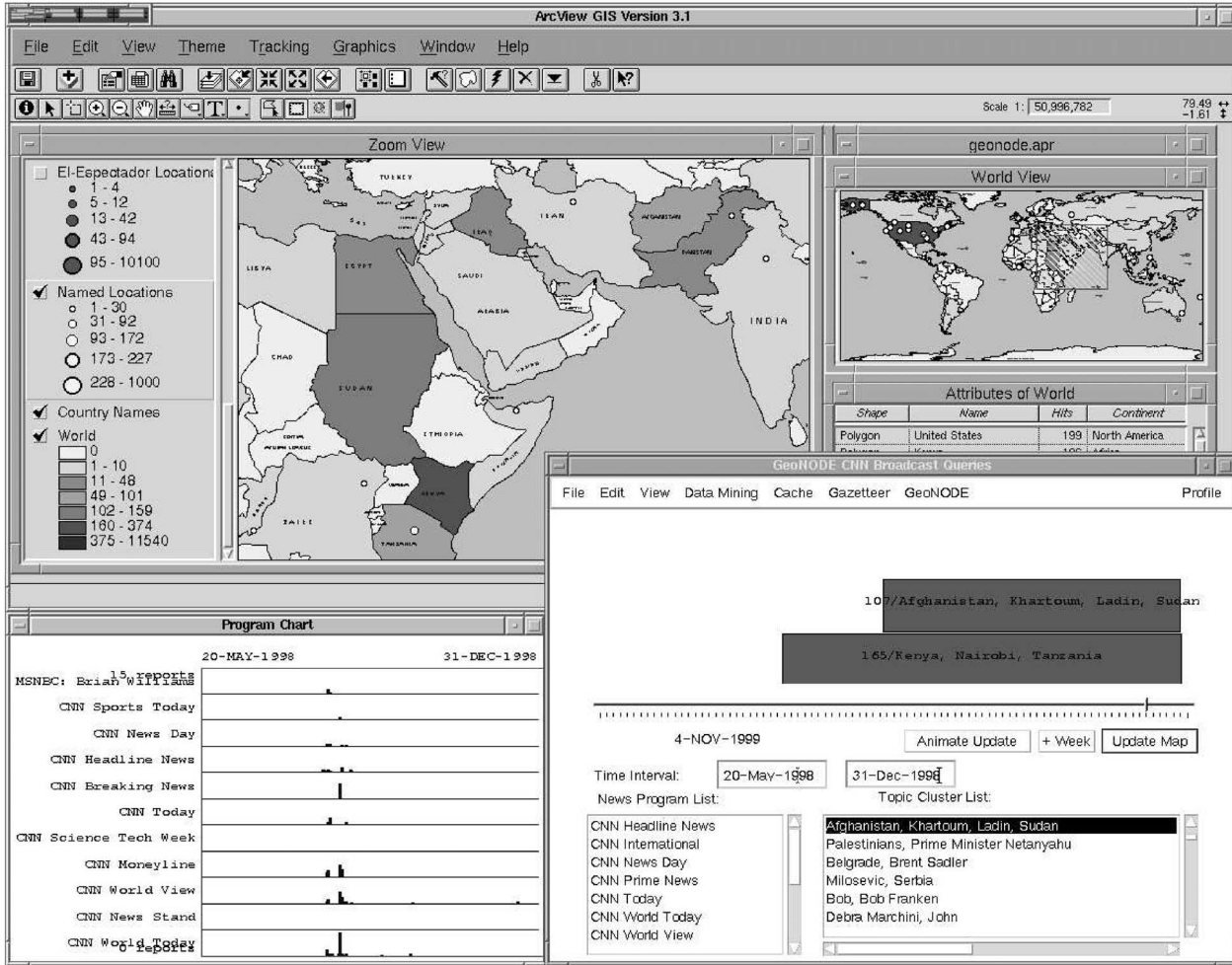
Fig. 1. GeoNODE screen shot showing identified topics at lower right.

2   PROBLEM STATEMENT AND RELATED WORK

The TopCat project started with a specific user need. The GeoNODE project at MITRE is developing a system for making news available to analysts [11]. One goal is to visualize ongoing topics in a geographic context; this requires *identifying* ongoing topics (see Fig. 1). We had experience with identifying association rules among entities/concepts in text, and noticed that *some* of the rules were recognizable as belonging to major news topics. This led to a topic identification mechanism based on data mining techniques.

Related problems are being addressed. The Topic Detection and Tracking (TDT) program [10] looks at two specific problems:

- **Topic Tracking:** Classify incoming documents into a predefined set of topics, based on a manually classified training set.
- **Topic Detection:** Recognize if a new document falls into an existing topic, or belongs in a new topic.

Our problem is similar to the Topic Detection (clustering) problem, except that:

- We must generate a *human-understandable* label for a topic: a compact identifier that allows a person to quickly see what the topic is about.

- Topic identification can be *retrospective*. We do not face the TDT requirement to identify each new document/ topic within a limited time after it arrives.

Although our goals are different, the test corpus developed for the TDT project provides a means to evaluate our work. The TDT corpus is a collection of news articles from the spring of 1998 and a ground truth topic set with documents manually classified into those topics. More discussion of the corpus and evaluation criteria is given in Section 4. The TDT2 [10] evaluation requires that we go beyond identifying topics and also match documents to a topic.

We thus define the topic identification problem as follows:

Definitions: Data Source:
Document: word+   *Corpus: {Document}*
TopicID: word+
Goal: Produce the following functions
TopicList(Corpus): {TopicID}
Topicmatch(TopicList(Corpus), Document ∈ Corpus):
   TopicID ⊂ TopicList(Corpus)

In Section 4, we show how to evaluate this problem using the TDT criteria and give an evaluation of TopCat's performance.

One key item missing from the TDT2 evaluation criteria is that the $TopicID$ must be *useful to a human*. This is hard to evaluate. Not only is it subjective, but there are many notions of useful. We will argue that the $TopicID$ produced by TopCat is useful to and understandable by a human.

Natural language processing and term clustering have both been employed in the Information Retrieval (IR) domain, usually to improve precision and recall [12], [13], [14]. Natural language processing has been used to automatically generate concept thesauri, generate document summaries, handle natural language queries, and reduce the feature space for vector space models, as discussed in [15]. A review of both statistical and natural language techniques for term extraction is given in [16]. Term clustering has also been used for automatic thesaurus generation, as well as for document clustering [17]. However, these techniques have rarely been used to understand a *collection*, as opposed to individual documents. There has been work in visualization of document collections (e.g., SPIRE [18]); however, these show relationships among keywords rather than identifying topics.

Perhaps the closest approach to this problem, other than the Topic Detection and Tracking work mentioned above, has been clustering of web search results. Such systems have similar goals to ours, such as performance and developing a human-browsable identifier. There are two key differences. The first is the Web as a data source. This provides information such as links and Web site addresses that can be used as clustering and naming criteria [19], [20]—many of our sources do not have these. The second, and perhaps more critical, difference is that these systems *start* with a focused search, as opposed to a broad-based corpus. The use of recurring phrases, as in Grouper [21], would seem less applicable with a broader corpus.

3 PROCESS

TopCat employs a multistage process, first identifying key concepts within a document, then grouping these to find topics, and finally mapping documents to topics and using the mapping to find higher-level groupings. Fig. 2 gives an overview. Natural language techniques extract named people, places, and organizations; identifying key concepts within a document. This gives a structure that can be mapped into *market basket* mining.[1] We then generate *frequent itemsets*, or groups of named entities that often appear together. Further clustering, using a hypergraph splitting technique, finds groups of frequent itemsets with considerable overlap.

The generated topics, a set of named entities, are used as a query to find documents related to the topic (Section 3.4). Where documents map to multiple topics, we perform a further clustering step that both joins similar topics and identifies topic/subtopic relationships.

Throughout this section, we will give examples and numbers based on the full six month TDT2 data set. We will use the following cluster, capturing professional tennis stories, as a running example:

PERSON	Andre Agassi	PERSON	Martina Hingis
PERSON	Mary Pierce	PERSON	Pete Sampras
PERSON	Serena	PERSON	Venus Williams
PERSON	Marcelo Rios	PERSON	Anna Kournikova

This is a typical cluster (in terms of size, support, etc.) and allows us to illustrate many of the details of the TopCat process. This cluster results from merging two subsidiary clusters (described in Section 3.5, formed from clustering seven frequent itemsets (Section 3.3).

3.1 Data Preparation

TopCat first uses Alembic [22] to identify *named entities* in each article. Alembic uses linguistic cues to identify people, places, and organizations in the text.[2] This shrinks the data set for further processing. It gives structure to the data; treating documents as a set of typed and named entities gives the information a schema suited to the market basket data mining problem. Third, and most important, from the start we are working with data that is rich in meaning, improving our chances of getting human understandable results.

Note that the use of named entities, as opposed to full text, is debatable. It has been shown that careful feature selection only slightly improves results in text categorization, while poor feature selection can have a large negative impact [23]. This leaves the question, are named entities a *good* form of feature selection?

We tested this on our data set using Support Vector Machines as classifiers [24]. Using the TDT2 training/ development sets as our training and test sets (stemmed using the Porter stemming algorithm [25] and filtered for a list of common stopwords), we obtained a precision of 95 percent for full text categorization versus 82 percent for named entity-based categorization (the recall was nearly identical, at 87 percent and 86 percent, respectively): Full text was better than named entities. Details of this test are given in [26].

However, for topic *identification*, the superiority of full text is not nearly as clear. We tested TopCat with full text and found two problems. The first was with computation time. The stemmed/filtered full text corpus contained almost 5 million unique word-document pairs versus 385,420 named entity/document pairs. On our prototype, we were unable to generate frequent itemsets at the low levels of support we used with named entities (at 5 percent support, it took nine hours on full text and only a single two-itemset was found.) We tried a smaller test set (one week of data) and the TopCat process took approximately one hour at 2 percent support. Using named entities from the same data took only two minutes at 0.5 percent support.

More critical is the difference in the *quality* of the results. With 2 percent support, operating on full-text generated 91 topics. Many were nonsensical, such as (tip, true) and (chat, signal, insid[3]), or nontopic relationships such as (husband, wife). The named entities, even at lower support, generated only 33 topics for the week and none were nonsensical (although some, such as (Brussels, Belgium), were not good topics). Even the best full-text clusters were not that good; Table 1 shows the *Asian Economic Crisis* cluster from the full-text and named-entity versions. We

---

1. Treating a document as a basket of words did not work well, as shown in Section 3.1. Named entities stand alone, but raw words need sequence to be meaningful.

2. Although not tested specifically on the TDT2 corpus, Alembic and other top Named Entity tagging systems typically achieve 90-95 percent precision and recall.
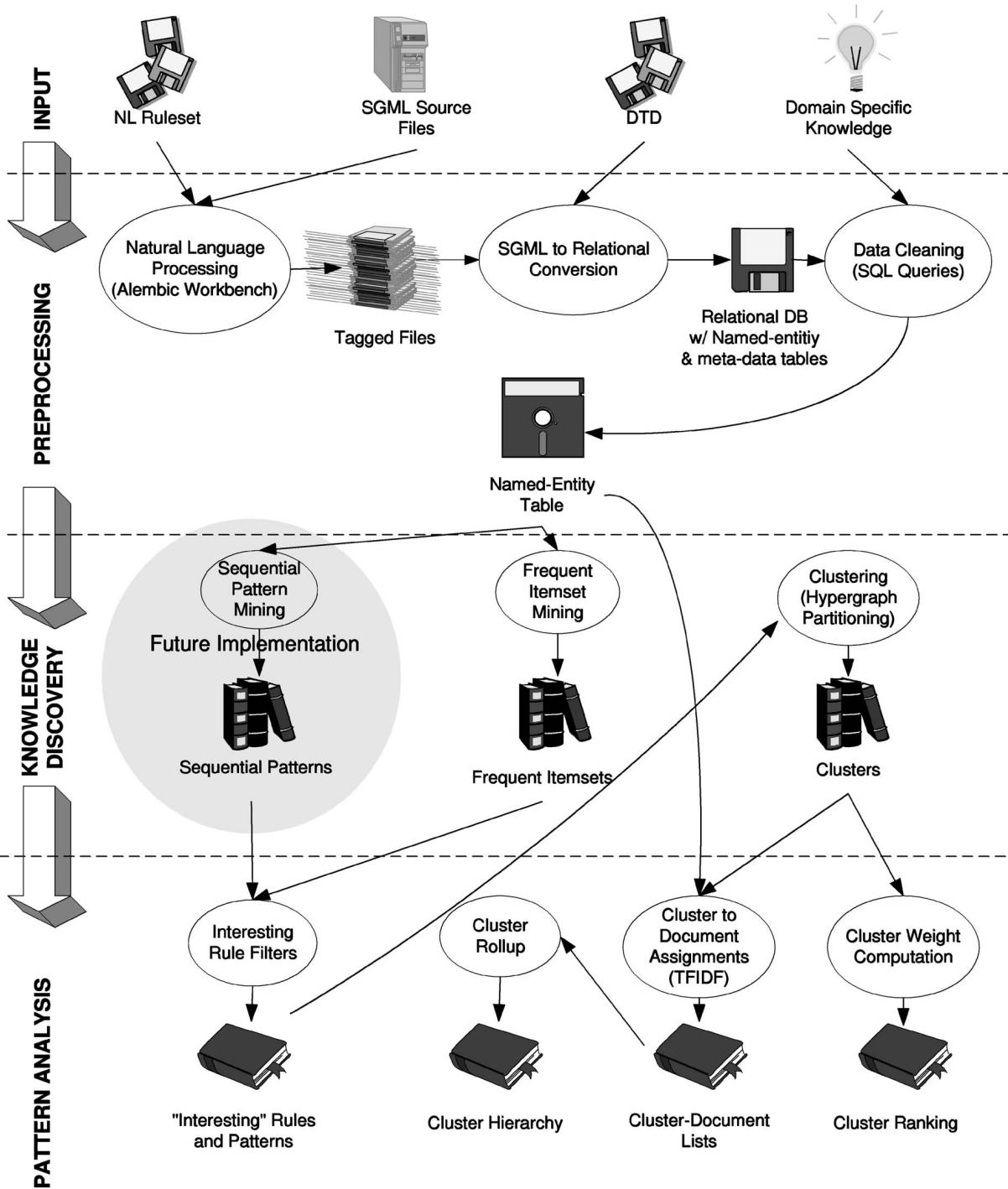
3. Note the use of stemmed words.

Fig. 2. TopCat process.

feel the named entity topic is as recognizable and gives more *useful* information. A domain-specific keyword set gives some improvement, as described in Section 3.1.2.

3.1.1 Coreference

One difficulty with named entities is that multiple names may be used for a single entity. This gives us a high correlation between different variants of a name (e.g., Rios and Marcelo Rios) that add no useful information. We want to capture that these all refer to the same entity, mapping multiple instances to the same variant of the name, before proceeding.

There are two issues involved:

1. How do we identify multiple references to the same entity within a document and
2. How do we ensure that the same name is used to refer to an entity *between* documents?

We have tried two approaches. The first is to find association rules where the predicted item is a substring of the predictor. This is used to build a global translation

TABLE 1
Asian Economic Crisis Topic: Full Text versus
Named Entities from One Week of News

Full Text	Named Entity	
analyst	LOCATION	Asia
asia	LOCATION	Japan
thailand	PERSON	Suharto
korea	LOCATION	China
invest	ORGANIZATION	International Monetary Fund
growth	LOCATION	Thailand
indonesia	LOCATION	Singapore
currenc	LOCATION	Hong Kong
investor	LOCATION	Indonesia
stock	LOCATION	Malaysia
asian	LOCATION	South Korea
	ORGANIZATION	Imf

table, changing all occurrences of the substring to the long version. This works well for names where the abbreviated variant name is uncommon (e.g., organization abbreviations), but is less effective with person names.

The second approach makes use of natural language techniques that work within a document. We use coreference information generated by Alembic to generate groups of names within a document that refer to the same entity (solving Problem 1 above). We still face Problem 2, however. Choosing the most common version isn't the right solution. (e.g., Marcelo Rios is referred to as Marcelo Rios 82 times and Rios 264, but there are 73 references to Rios that refer to someone else). Using the least common variant is also a poor choice, as many documents may not contain that variant (exacerbating Problem 2). Our solution is to use the globally most common version of the name *where most groups containing that name contain at least one other name within the current group.* Although not perfect (e.g., three documents referencing Marcelo Rios only as Rios are missed), this does give a global identifier for an entity that is both reasonably global and reasonably unique.

In many cases, this is better than such obvious techniques as using a full name. For example, Serena Williams is referred to simply as Serena in many articles (the full name is never mentioned); the above technique captures this in choosing a global identifier. More sophisticated techniques could be used, such as a manually prepared catalog of global names, but we find this sufficient for our purposes.

Although the natural language technique is our primary approach, we also use the association rule-based approach with a minimum support of 0.05 percent and a minimum confidence of 50 percent. This produces six additional translations.

3.1.2 Keywords

Named entities capture "Who?" and "Where?" (and date tagging exists to capture "When?"), but require that we use our background knowledge to understand "What?" and "Why?" As we have seen, full text gives a flood of often irrelevant information. Another possibility is human-generated keywords. By generating a set of keywords that

capture concepts of interest, we can extend the concepts used in topic identification at constant human cost.

Obtaining a good set of keywords is a difficult task. To keep the human cost small, we do not require human generation of a comprehensive keyword set. Instead, we use WordNet [27] to automatically expand the keyword list to cover the concept. WordNet is a semantic network that forms a hierarchical lexicon of 100,000 word forms. It includes synonyms, antonyms, and hierarchical relations: hypernyms and hyponyms. A hypernym is a word that is more general than another word, a hyponym is a word that is more specific. For example, vehicle is a hypernym of automobile and couch is a hyponym of furniture. The WordNet hyper/hyponym relations form a set of directed acyclic graphs (DAGs). We define the depth of root words to be 1 and any other word to be one plus the depth of its shallowest hypernym/hyponym. We qualitatively evaluated hypernyms and hyponyms of each word in the topic statement for 20 Text REtrieval Conference queries [28] for relevance. At depth 5 and greater, the hypernyms represented similar concepts. Wide semantic jumps with hyponyms tended to occur when a word has many hyponyms; we found that the hyponyms of words with 15 or fewer hyponyms avoided large semantic leaps. By exploiting these relations, we expand a set of keywords to include related words describing the same concept.

We have developed the following three heuristics for controlling the aspects of WordNet that should be used in keyword expansion:

1. A (word, sense) pair given by a WordNet relation should be added to the expanded keyword list only if the sense is the most common one for that word.
2. A hypernym relation should be used only if the hypernym is at depth 5 or below.
3. A hyponym relation should be used only if there are no more than 15 hyponyms for the corresponding keyword.

These heuristics give a set of rules that produce a fairly robust keyword set. For example, given the keyword set president, U.S., keyword expansion yields President of the United States, President, Chief Executive, head of state, chief of state, United States, United States of America, America, US, USA, U.S.A., North American country, North American nation, a significant improvement in breadth.

We tested keywords with TopCat using four "concepts" and keywords sets:

DISASTERS:	accident, avalanche, death, disaster, earthquake, tornado
TRIALS:	court, lawsuit, lawyer, suit, trial
US_POLITICS:	President, U.S., democrat, election, legislation, republican
VIOLENCE:	bomb, hostage, protest, raid, violence

Keyword expansion gave 137 total keywords from this initial set of 22. In practice, we would expect the concepts to be more tightly defined. With few occurrences of keywords in a group, we could treat the keywords in a group as identical (we have not done so with these groups as they exceed the 5 percent stop limit defined below.) This would
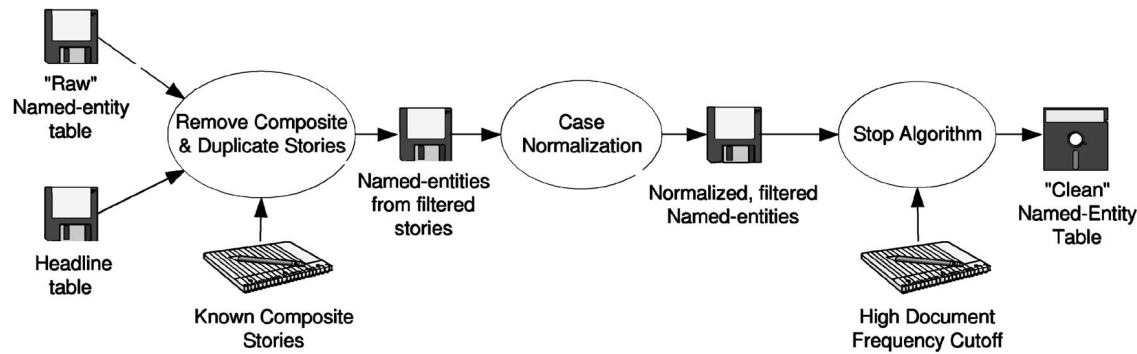
Fig. 3. TopCat data cleaning.

help to customize the topics to the needs of the user, as well as improve the clarity of the discovered topics. In Section 4, we will discuss the relative performance of TopCat with and without the addition of keywords.

3.1.3 Data Cleansing

Several data cleaning algorithms are applied to increase the quality of the results, as shown in Fig. 3. The generic data cleaning techniques include case normalization and a stop algorithm. Named entity identification eliminates words traditionally found in a stop list. Our stop algorithm removes terms occurring in over 5 percent of the articles as these are used in too many topics to effectively discriminate between topics. The idea that frequency is inversely proportional to the usefulness of a term is commonly accepted (e.g., Salton's TFIDF (term frequency/ inverse document frequency) term weighting scheme [29] for Information Retrieval, see footnote 6.) This eliminates only a few entities—in the TDT2 evaluation, United States and Clinton. Although potentially confusing (note the lack of United States in the Iraq/UN cluster in Table 7), it becomes unnoticeable with use and results in more concise topic identifiers.

TopCat also uses *corpus specific* data cleaning steps: removal of duplicate stories (an artifact of pulling stories from a newswire, where errors cause the entire story to be retransmitted) and removal of what we refer to as *composite* stories. A composite story is a multitopic story that contains brief descriptions or recaps of stories reported elsewhere. In print media, composite stories often appear on the first page of a section, with brief descriptions of stories contained within the section or stories that have occurred in the previous week. If these stories are not filtered out before the knowledge discovery phase, terms and stories are associated with each other simply because the events are reported in the same section of the newspaper or occur over the same time period. A *composite* story is different from a simple multitopic story as the topics covered in a composite story are generally covered elsewhere in the paper. The heuristic TopCat uses for identifying composite stories is to look for reoccurring identical headlines. Any headline that occurs on at least a monthly basis (e.g., BULLETIN) is assumed to be a composite story and is filtered out.

3.2 Frequent Itemsets

The foundation of the topic identification process is *frequent itemsets*. In TopCat, a frequent itemset is a group of named entities that occur together in multiple articles. Cooccurrence of words has been shown to carry useful information [30], [31], [32]. What this information really gives us is correlated items, rather than a topic. However, we found that correlated named entities frequently occurred within a recognizable topic—clustering the interesting correlations enabled us to identify a topic. Before going into the clustering method in Section 3.3, we will first describe how to determine *interesting* correlations.

Discovery of frequent itemsets is a well-understood data mining problem, arising in the *market basket* association rule problem [33]. A document can be viewed as a market basket of named entities; existing research in this area applies directly to our problem. The search is performed directly in a relational database using *query flocks* [34] technology, allowing us to incorporate the filtering criteria described below into the search while relying on the database query processor for many algorithmic issues. The computational complexity is essentially that of the Apriori algorithm [35]. Apriori grows linearly with the number of transactions (documents) and the number of *candidate* itemsets. The problem is, the number of candidate itemsets is potentially exponential in the number of items (named entities). Setting a high threshold on the support (frequency of cooccurrence) decreases the number of candidate itemsets. Agrawal and Srikant obtained roughly linear increase in execution time as support decreased. Our results generally agree with this—although, below a certain point, the number of itemsets does increase exponentially. This occurred because the corpus had duplicate (or near duplicate) documents, such as multiple news stories based on the same newswire article. Each duplicate document set gives a very large itemset, with a combinatorial explosion in the number of small itemsets that occur in that large itemset.

The use of support as a threshold causes TopCat to ignore topics that occur in few documents. This fits well with the original goal of the system. The TDT2 corpus used many smaller topics, however, so we did test TopCat to see how it would perform with low support thresholds. We found that a threshold of 0.06 percent (30 documents in the TDT corpus) gave reasonable results on the TDT2 training data, as well as performing well with other corpora.

Since we are working with multiple sources, any topic of importance is mentioned multiple times; this level of support captures all topics of any ongoing significance. However, this gives a total of 21,173 frequent itemsets, of which 6,028 were 2-itemsets, and most of the rest were 3 and 4 itemsets. There were a few larger itemsets, with the largest being an 11-itemset coming from the UN Security Council/Iraq Arms inspections topic. Although the largest itemsets were interesting, many of the smaller ones were not important. We need additional filtering criteria to get just the important itemsets.[4]

3.2.1 Filtering of Frequent Itemsets

The traditional market basket association rule filters are:

support   the number (or percent) of baskets that must contain the given rule (also known as cooccurrence frequency); and

confidence   the percent of time the rule is true (given the antecedent, the consequent follows).

We have already discussed problems with support; although useful, it is not sufficient as a filtering mechanism. Confidence overemphasizes common items as consequents and rare items as antecedents (e.g., "Key West $\Longrightarrow$ United States"). The consequent in such cases rarely adds much meaning to a topic identifier.

Instead of confidence, we use mutual information [36]:

$$\log_2 \frac{P(x,y)}{P(x)P(y)}.$$

This is a measure of correlation strength, i.e., the ratio of the actual probability of a frequent itemset occurring in a document to the probability of the items occurring together by chance. This measure emphasizes relatively rare items that generally occur together and deemphasizes common items. Mutual information has been shown to be an effective word association norm; it is basically the same as the *interest* measure used for text associations rules in [32] and similar to the *association ratio* of [30] used for words occurring in close proximity.

We use both support and mutual information. Very high support itemsets are almost always significant, as are high mutual information itemsets. We select all frequent itemsets where either the support or mutual information is at least one standard deviation above the average for that metric or where both support and mutual information are above average. The average and standard deviation are computed independently for 2-itemsets, 3-itemsets, etc. For 2-itemsets, this brings us from 6,028 to 1,033 and brings the total from 21,173 to 3,072. This is still dependent on the choice of a minimum support; computing the averages *efficiently* without a fixed minimum support is an interesting problem.

We also use mutual information to choose between "contained" and "containing" itemsets (e.g., any 3-itemset contains three 2-itemsets with the required support.) Since the information in the contained itemsets is represented in the containing itemset, we can eliminate them. However, a strong 2-itemset may be more meaningful than a weak 3-itemset. An

TABLE 2
Individual Matches of Significance

Value1	Value2	Support	Mutual Information
Andre Agassi	Marcelo Rios	.00063	8.0
Andre Agassi	Pete Sampras	.00100	7.6
Anna Kournikova	Martina Hingis	.00070	8.1
Marcelo Rios	Pete Sampras	.00076	8.0
Martina Hingis	Mary Pierce	.00057	7.8
Martina Hingis	Serena	.00054	7.8
Martina Hingis	Venus Williams	.00063	7.5

$(n-1)$-itemset is kept only if it has greater mutual information than the corresponding n-itemset and an n-itemset is used only if it has greater mutual information than at least one of its contained $(n-1)$-itemsets. This filter brings us to 416 (instead of 1,033) 2-itemsets, with even greater reduction among the larger itemsets (for example, all of the 10-itemsets were contained in the 11-itemset). Overall, this reduced the number of frequent itemsets to 865.

A problem with using frequent itemsets for topic identification is that they tend to be over-specific. For example, the tennis player frequent itemsets consist of those shown in Table 2. These capture individual matches of significance, but not the topic of *championship tennis* as a whole. There are also some rules containing these players that are filtered out due to low support and/or mutual information, such as locations of matches and home countries of players (interesting, perhaps, but not relevant to the overall topic).

3.3 Clustering

We experimented with different frequent itemset filtering techniques, but always found an unacceptable trade off between the number of itemsets and the breadth of topics covered. Further investigation showed that some named entities that should be grouped as a topic would not show up as a frequent itemset under *any* measure; no article contained **all** of the entities. Therefore, we chose to perform clustering of the named entities grouped by the frequent itemsets. We use a hypergraph-based method, based on that of [9].[5] We treat the frequent itemsets as edges in a hypergraph, with named entities as the nodes. We repeatedly partition the hypergraph; the remaining connected graphs give the named entities in a topic.

Clustering based on the partitioning of a frequent itemset hypergraph was chosen for two reasons. First, the method easily handles the large number of dimensions associated with the text domain. Second, the method is efficient given that we have already found frequent itemsets. The *hypergraph clustering* method of [9] takes a set of association rules and declares the items in the rules to be vertices and the rules themselves to be hyperedges. Since association rules have a directionality associated with each rule, the algorithm combines all rules with the same set of items and uses an average of the confidence of the individual rules as the weight

---

4. The problems with traditional data mining measures for use with text corpuses have been noted elsewhere as well. See [31] for another approach.

5. There have been other methods proposed for clustering frequent itemsets. A method based on *large items* (those appearing frequently in itemsets) [37] was considered, but was inappropriate for our problem as it concentrated clusters around common named entities that appear in multiple human-defined topics.

for a hyperedge. Clusters can be quickly found by using a hypergraph partitioning algorithm such as hMETIS [38].

We adapted the hypergraph clustering algorithm described in [9] in several ways to fit our particular domain. Because TopCat discovers frequent itemsets instead of association rules, the rules are not directional and do not need to be combined to form undirected edges in a hypergraph. The mutual information of each itemset was used for the weight of each edge.

Upon investigation, we found that the stopping criteria presented in [9] only works for very highly connected hypergraphs. Their algorithm continues to recursively partition a hypergraph until the weight of the edges cut compared to the weight of the edges left in either partition falls below a set ratio (referred to as *fitness*). This criteria has two fundamental problems:

- It will never divide a loosely connected hypergraph into the appropriate number of clusters as it stops *as soon as* it finds a partition that meets the fitness criteria; and
- It always performs at least one partition (even if the entire hypergraph should be left together). This can inappropriately partition a group of items that should be left together. If the initial hypergraph is a group of items that logically belong to a single cluster, the algorithm will go ahead and partition the items anyway.

To solve these problems and to allow items to appear in multiple clusters, we modified the algorithm as follows:

- hMETIS tries to split the hypergraph into two relatively equal parts while minimizing the weight of the edges cut. It will allow the number of vertices in each split to be unequal up to a given unbalance factor as long as this results in a lower cut weight. Our algorithm allows hMETIS to use as high an unbalance factor as necessary, with the restriction that the smallest partition size possible is two vertices. (A cluster of one item is rarely meaningful.) The algorithm automatically adjusts the unbalance factor based on the size of the hypergraph to allow for the maximum unbalance. This prevents a bad split from being made simply to preserve equal partition sizes.
- A *cutoff* parameter is used that represents the maximum allowable cut-weight ratio (the weight of the cut edges divided by the weight of the uncut edges in a given partition). The cut-weight ratio is defined as follows. Let P be a partition with a set e of m edges and let c be the set of n edges cut in the previous split of the hypergraph:

$$cutweight(P) = \frac{\Sigma_{i=1}^{n} Weight(c_i)}{\Sigma_{j=1}^{m} Weight(e_j)}.$$

A hyperedge is counted in the weight of a partition if two or more vertices from the original hyperedge are in the partition. For example, a cut-weight ratio of 0.5 means that the weight of the cut edges is half the weight of the remaining edges. The algorithm assumes that natural clusters will be highly connected by edges. Therefore, a low cut-weight ratio indicates that hMETIS made what should be a natural split between the vertices in the hypergraph. A high cut-weight ratio indicates that the hypergraph was a natural cluster of items and should not have been split.

- Once the stopping criteria has been reached for all of the partitions of a hypergraph, vertices are added back to clusters depending on the *minimum-overlap* parameter. Up to this point in the algorithm, a given vertex can only be a member of one cluster. Often, there are vertices that could logically belong to several clusters. For each partial edge that is left in a cluster, if the percentage of vertices from the original edge that are still in the cluster exceed the minimum-overlap percentage, the removed vertices are added back in. Overlap for an edge is calculated as follows, where v is the set of vertices:

$$overlap(e, P) = \frac{|\{v \in P\} \cup \{v \in e\}|}{|\{v \in e\}|}.$$

For example, if the minimum-overlap is set to 50 percent, and three of the original four vertices of an edge end up in the same cluster, the fourth vertex is added back in since the overlap for the edge is calculated to be 0.75. Once this is done, a check is made to remove any clusters that are a pure subset of another cluster (this often occurs with small clusters whose vertices are from an edge that is also part of a larger cluster).

Based on the TDT training and test data, we chose a cutoff ratio of 0.4, and a minimum-overlap ratio of 0.6.

Fig. 4 shows the hypergraphs created from the tennis player frequent itemsets. In this example, each hypergraph becomes a single cluster. Cuts *are* performed before the stopping criteria is reached, for example the Agassi/Sampras and Agassi/Rios links are cut. However, they are added back in the final step.

This produces the following two clusters:

PERSON	Andre Agassi	PERSON	Martina Hingis
PERSON	Pete Sampras	PERSON	Venus Williams
PERSON	Marcelo Rios	PERSON	Anna Kournikova
		PERSON	Mary Pierce
		PERSON	Serena

The TDT data produces one huge hypergraph containing half the clusters and several independent hypergraphs. Most of the small hypergraphs not partitioned. One that does become multiple clusters is shown in Fig. 5. Here, the link between Joe Torre and George Steinbrenner (shown dashed) is cut. This is not the weakest link, but the attempt to balance the graphs causes this link to be cut rather than producing a singleton set.

This is a sensible distinction. For those that don't follow US baseball, in 1998, George Steinbrenner owned and Joe Torre managed the New York Yankees. Darryl Strawberry and David Cone were star players. Tampa, Florida is where the Yankees train in the spring. During the January to April time frame, the players and manager were in Tampa training, but George Steinbrenner had to deal with repairs
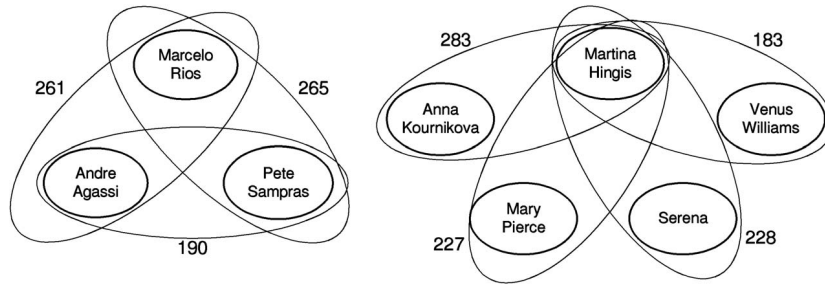
Fig. 4. Hypergraph of tennis player frequent itemsets.

to a crumbling Yankee Stadium back in New York—thus, the end result *does* reflect what really happened.

3.4 Mapping to Documents

The preceding process gives us reasonable topics. However, the original goal of supporting analysis of news requires allowing analysts to drill down from a topic to the stories making up that topic. We could trace back to the source data, tracking each frequent itemsets directly to its set of supporting documents. This has two problems:

1. A document can be responsible for multiple frequent itemsets for evaluating against the TDT2 criteria we need to identify a single topic for each document and
2. A document may relate to a topic, but not contain the *all* the entities of any of the frequent itemsets.

We instead use the fact that the topic itself, a set of named entities, looks much like a Boolean query. We use TFIDF[6] as a distance measure between a document and a topic, then choose the closest topic for each document. (In practice, we also use cutoffs when a document isn't close to any topic and allow multiple mappings if it is close to many.) Note that this calculated with named entities; we need not refer to the full text.

3.5 Combining Clusters Based on Document Mapping

Although the clustered topics appeared reasonable, the segments were too fine-grained with respect to the TDT human-selected topics. For example, we separated men's and women's tennis; the TDT human-defined topics had this as a single topic.

We found that the topic-to-document mapping provided a means to deal with this. Many documents were close to multiple topics. In some cases, this overlap was common and repeated; many documents referenced both topics (the tennis example was one of these). We used this to merge topics, giving a final tennis topic of:

PERSON Andre Agassi
PERSON Martina Hingis
PERSON Mary Pierce

6. The TFIDF weight between a document i and topic t is calculated as follows: [29]

$$TFIDF_{it} = \sum_{k \in t} \frac{tf_{ik} \cdot (\log(N/n_k))^2}{\sqrt{\sum_{j \in t} (\log(N/n_j))^2} \sqrt{\sum_{j \in t} (tf_{ij})^2 (\log(N/n_j))^2}},$$

where tf_{ik} is the term frequency (number of occurrences) of term k in i, N is the size of the corpus, and n_k is the number of documents with term k.

PERSON Pete Sampras
PERSON Venus Williams
PERSON Serena
PERSON Marcelo Rios
PERSON Anna Kournikova

These relationships capture two different types of overlap between topics. In the first, *marriage*, the majority of documents similar to either topic are similar to both. In the second, *parent/child*, the documents similar to the child are also similar to the parent, but the reverse does not necessarily hold. The tennis clusters were a *marriage* merge. A graphic description of the types of relationships is given in Fig. 6. The calculation of these values is somewhat more complex as it also uses negative relationships.

3.5.1 Marriage Relationship Calculation

The marriage similarity between clusters a and b is defined as the average of the product of the TFIDF scores for each document across the clusters, divided by the product of the average TFIDF score for each cluster:

$$Marriage_{ab} = \frac{\frac{\sum_{i \in documents} TFIDF_{ia} * TFIDF_{ib}}{N}}{\frac{\sum_{i \in documents} TFIDF_{ia}}{N} \frac{\sum_{i \in documents} TFIDF_{ib}}{N}}.$$

This is again a mutual information style of metric. Intuitively, if a document is in a and not in b, that document contributes 0 to the sum in the numerator—if the clusters have no overlap, the numerator is 0. Since the TFIDF measure is continuous, it is more complex. Basically, if a document is similar to a and b, it contributes to the numerator and, if it is dissimilar to both, it doesn't contribute to the denominator. If the TFIDF values were assigned randomly (no particular correlation, either positive or negative, between a and b), the expected value for
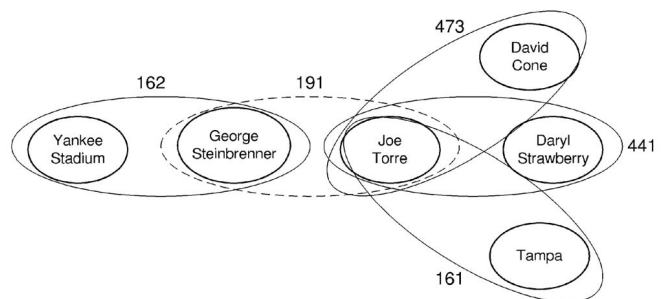

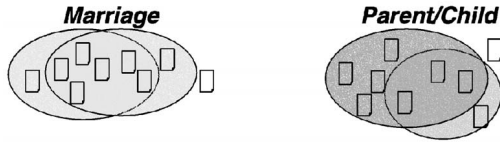Fig. 5. Hypergraph of New York Yankees baseball frequent itemsets.

Fig. 6. Types of relationships.

$Marriage_{ab}$ would be 1. Values less than 1 imply a negative correlation between the clusters.

Based on experiments on the TDT2 training set, we chose a cutoff of $Marriage_{ab} \geq 30$ for merging clusters. This is not a transitive measure; this could pose a problem where clusters a and b are marriages, b and c are marriages, but a and c are not. However, to merge clusters, we take a union of their named entities, not the related documents. Since topic identifiers need not partition the set of entities, the lack of transitivity is not a practical issue (we get two topics from the original three). We merge into a single cluster where such transitivity exists.

We had 47 pairs with similarity greater than 30 for the marriage relationship in the TDT data. The two examples with highest similarity are given in Table 3. Most consisted of two topics; however, one each contained three, five, and six topics; reducing the total number of topics by 36. The largest of these merges the various weather forecasters (originally individual topics) into the single group shown in Table 7.

3.5.2 Parent/Child Relationship Calculation

The parent/child relationship is similar, but nonsymmetric. It captures the relative similarity between child documents and the parent. For example, if a is a large cluster and b is small, they would not be similar under $Marriage_{ab}$ as the

first term in the denominator would dominate. The parent/child relationship similarity is calculated as follows:

$$ParentChild_{pc} = \frac{\sum_{i \in documents} \frac{TFIDF_{ip} * TFIDF_{ic}}{N}}{\sum_{i \in documents} \frac{TFIDF_{ic}}{N}} .$$

This metric ranges from 0 to 1, with a value of 1 indicating that everything in the child is contained in the parent. We calculate the parent/child relationship after the marriage clusters have been merged. Merging the groups is again done by a union of the named entities.

The Parent/Child relationship gave 16 pairs with a similarity greater than 0.3 in the TDT data. These are divided into seven hierarchies. Many marriage and parent/child relationships overlapped; seven parent/child pairs remained after merging with the marriage relationship. The three highest similarity pairs (note that the India/Pakistan topic has two children) are given in Table 4.

Note that there is nothing *document*-specific about these methods. The same approach could be applied to any market basket problem.

3.6 Parameter Settings

TopCat has several parameters whose adjustment affects results (Table 5). The results are not that sensitive to changes in most parameters. We now discuss how the default values were chosen, effects of modifying those parameters, and suggestions for practical uses.

The first three parameters, used in data preparation, affect a very small number of items and can be checked manually. The frequent item cutoff eliminated only United States and Clinton in the TDT2 evaluation set. In the full TDT2 data set, Washington was also dropped. There were only five items with support between 4 and 5 percent. This cutoff eliminates items that are so frequent as to skew the

TABLE 3
Topics with Greatest Similarity under Marriage Relationship

Topic		Topic		Similarity
LOCATION	Dubai	ORGANIZATION	Crown	103
LOCATION	United Arab Emirates	PERSON	Abdullah	
ORGANIZATION	Mets	PERSON	Bernard Gilkey	204
PERSON	Valentine	PERSON	Carlos Baerga	

TABLE 4
Topics with Greatest Similarity under Parent/Child Relationship

Parent		Child		Similarity
ORGANIZATION	Congress	PERSON	Dick Gephardt	0.55
ORGANIZATION	White House	PERSON	Newt Gingrich	
ORGANIZATION	House			
PERSON	Newt Gingrich			
ORGANIZATION	Senate			
LOCATION	India	ORGANIZATION	Bjp	0.46
LOCATION	Islamabad	ORGANIZATION	Congress Party	
ORGANIZATION	Bjp			
LOCATION	New Delhi	LOCATION	Islamabad	0.42
LOCATION	Pakistan	PERSON	Nawaz Sharif	
LOCATION	South Asia			

TABLE 5
TopCat Parameters

Section	Parameter	Value used
3.1.1	Coreference support	0.05%
3.1.1	Coreference confidence	50%
3.1.3	Stop frequency (too frequent entities)	5%
3.2	Support for frequent itemsets	0.06%
3.2.1	Support only threshold	one standard deviation above average
3.2.1	Mutual information only threshold	one standard deviation above average
3.2.1	mutual information + support threshold	above average
3.3	Cutoff parameter	0.4
3.3	minimum overlap	0.6
3.5.1	marriage cutoff	30
3.5.2	parent/child cutoff	0.3

results and that contain little semantic information (bylines in news articles are a common example.) The name translation parameters produce a small number of items (six in the TDT data set), but, as they are frequent, it has a substantial impact on the results. Most are straightforward (e.g., sports team full names versus short names, such as New York Rangers versus Rangers); these are frequently abbreviated in short articles and are missed by the single document natural-language approach. The only questionable translation was Korea to South Korea; a sample of the documents affected showed this to be appropriate. While we have found no need to adjust these for other corpuses, a simple sanity check when moving to a new type of data is appropriate.

The support level and filtering criteria for frequent itemsets are perhaps the most difficult parameters. The filtering criteria were set empirically using the TREC Wall Street Journal data set and a continuously varying collection of broadcast news, and proved quite resilient to adjustments. They are self-adjusting as the support level and data set change. However, the evaluation was sensitive to changes in the support level. Topics that are discussed in a few stories disappear as the support level increases. While okay for many applications (e.g., a top 10 list of topics), it posed problems for the TDT2 test. However, at extremely low support levels, near-duplicate stories cause the number of frequent itemsets to explode. This is a particular problem with small data sets where near-duplicate stories are likely, e.g., identifying topics in the results of a query. We are currently working on dynamic approaches to setting minimum support based on the relative number of k and $k + 1$ itemsets.

Topic identification was quite insensitive to changes in the cutoff and minimum overlap parameters. For example, varying the cutoff from 0.4 to 0.6 produced 169 versus 177 topics. The added topics were of little significance. Varying the overlap from 0 to 0.65 (at cutoff 0.5) increased the number of items in the 175 topics from 453 to 521, and generated two additional topics.

The marriage and parent/child parameters had a significant effect on the TDT training data. The marriage cutoff of 30 was a reasonably clear choice—on the training

and test data sets, there were few topics with similarity in the range 25 to 35. The parent/child similarity also had a natural cutoff at 0.3; the highest similarity was 0.4 and the closest to 0.3 were 0.27 and 0.35. In practice, these steps are unnecessary as the combined topics generally make sense as independent topics. These steps are more useful to show the relationship between topics (see Fig. 7). However, they were needed to give the topic granularity required for the TDT2 training (but not evaluation) data, as discussed in Section 4.

4 EXPERIMENTAL RESULTS: TOPCAT VERSUS HUMAN-DEFINED TOPICS

Evaluating TopCat is difficult. The goal is to identify a topic that makes sense to a person, a subjective measure. The only large document corpus we are aware of with clearly defined topics is the Topic Detection and Tracking program [10]. This corpus contains January to June 1998 news from two newswires, two televised sources, and two radio sources. It has over 60,000 stories, the majority from the newswires. One
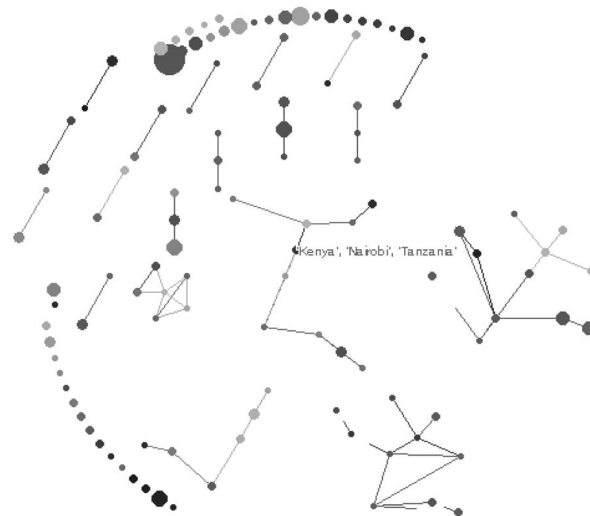

Fig. 7. Display of relationships found in broadcast news.

hundred topics were chosen and all stories on each topic were manually identified, covering about 10 percent of the corpus. An example topic is *Bombing AL Clinic*, about the 29 January 1998 bombing of a clinic in Alabama, and the following investigation. (TopCat identified this as "Alabama Birmingham, Eric Robert Rudolph"—Rudolph was a prime suspect.) Details on the construction of the corpus are given in [39]. Other commonly used corpuses, such as Reuters-21578 [40] or usenet newsgroups, do not define topics in a way that enables an objective topic discovery evaluation.

While comparing the TopCat-produced topic identifiers to the human-defined TDT2 labels would be subjective, the quality of topics can be measured by treating this as a clustering problem. The TDT2 program addressed clustering and classification of documents. Since clustering documents into topics (the *Topic Detection* task of TDT) enumerates topics, the human-generated TDT test corpus provides a useful testbed for TopCat. Each topic has a corresponding group of documents—comparing the cluster with the actual documents gives miss and false alarm ratios. The TDT2 program combines the probability of failing to retrieve a document that belongs with the topic (P_{Miss}) and the probability of erroneously matching a document to the topic ($P_{FalseAlarm}$) into a single cost of detection or C_{Detect} score [41]:

$$C_{Detect} = C_{Miss} \cdot P_{Miss} \cdot P_{topic} + C_{FalseAlarm} \cdot P_{FalseAlarm} \cdot (1 - P_{topic})$$

where:

$$P_{Miss} = \sum_R |R - H(R)| / \sum_R |R|$$
$$P_{FalseAlarm} = \sum_R |H(R) - R| / \sum_R |S - R|$$

R is the set of stories in a reference target topic.

$H(R)$ is the set of stories in the TopCat-produced topic best matching R.

P_{topic} $= 0.02$ (the a priori probability of a story in the corpus being on a given topic.)

C_{Miss} $= 1$ (the chosen cost of a miss.)

$C_{FalseAlarm}$ $= 1$ (the chosen cost of a false alarm.)

The TDT2 evaluation process specifies that the mapping $H(R)$ between TopCat-identified topics and reference topics be defined as the mapping that minimizes C_{Detect} for that topic. This is calculated as follows:

$$H(R) = \underset{H}{argmin}\{C_{Detect}(R, H)\}$$

$$C_{Detect}(R, H) = C_{Miss} \cdot P_{Miss}(R, H) \cdot P_{topic} + C_{FalseAlarm} \cdot P_{FalseAlarm}(R, H) \cdot (1 - P_{topic})$$

$$P_{Miss}(R, H) = N_{Miss}(R, H) / |R|$$
$$P_{FalseAlarm}(R, H) = N_{FalseAlarm}(R, H) / |S - R|$$

$N_{Miss}(R, H)$ = the number of stories in R that are not in H.

$N_{FalseAlarm}(R, H)$ = the number of stories in H that are not in R.

$|X|$ = the number of stories in the set X of stories.

S = the stories to be scored in the evaluation corpus being processed.

For the TDT2 competition, the corpus was divided into separate training, test (parameter setting), and evaluation data. Using the TDT2 evaluation data (May and June), the TopCat C_{Detect} score was 0.0062 using named entities alone, with improvements up to 0.0053 when a selection of keywords in the categories DISASTERS, TRIALS, VIOLENCE, and US_POLITICS were added (as described in Section 3.1.2). This was comparable to the results from the TDT2 topic detection participants [42], which ranged from 0.0040 to 0.0129. This shows that TopCat's performance at clustering is reasonable. We will discuss this in more detail in Section 4.1; however, first we give more discussion of the results of TopCat on the TDT2 corpus.

Of particular note is the low false alarm probability of TopCat (0.0021); further improvement here would be difficult. The primary impediment to a better overall score (contributing $\approx 2/3$ of the C_{Detect} score) is the miss probability of 0.19. Performance of TopCat on the entire six month TDT2 corpus was substantially lower—a C_{Detect} score of 0.011. The false alarm probability stayed similar (0.0026), but the miss ratio went to 0.42. The TDT2 participants experienced similar results—this is primarily due to several large, evolving topics that were in the training and test sets but not part of the evaluation criteria.

The main reason for the high miss probability is the difference in specificity between the human-defined topics and the TopCat-discovered topics. (Only three topics were missed entirely; containing one, three, and five documents.) Many TDT2-defined topics corresponded to multiple TopCat topics. Since the TDT2 evaluation process only allows a single system-defined topic to be mapped to the human-defined topic, over half the TopCat-discovered topics were not used, and any document associated with those topics was counted as a "miss" in the scoring. In testing against the full six months of data, over half of the misses were associated with three big topics: The East Asian economic crisis, the problems in Iraq, and the 1998 Winter Olympics. TopCat often identified separate topics corresponding to the human-selected TDT2 topic. For example, TopCat identified both an overall Iraq conflict topic (shown later at the top of Table 7), as well as a US specific topic of Madeleine Albright/Iraq/Middle East/State. The East Asian economic crisis was even more significant, with TopCat identifying topics such as Jakarta/Suharto (Indonesia) and IMF/International Monetary Fund/Michel Camdessus in addition to the following "best" topic (lowest C_{Detect} score):

LOCATION	Asia
LOCATION	Indonesia
LOCATION	Japan
LOCATION	Malaysia
LOCATION	Singapore
LOCATION	South Korea
LOCATION	Thailand

This is the best *Asian economic crisis* topic, but it has a miss probability of 0.61. Including all 14 TopCat topics that match the Asian economic crisis better than any other topic would lower the miss probability to 0.22. Although various TopCat parameters could be changed to merge these, many topics that the ground truth set considers separate (such as the world ice skating championships and the winter Olympics) would be merged as well.

The TFIDF-based topic merging of Section 3.5 addressed this, substantially improving results in the training set. Interestingly, topic merging did not have a significant effect on the evaluation—without it, TopCat would have had $C_{Detect} = 0.0061$. This results from the way the evaluation set was constructed: The evaluation set did not include

topics found in the training and test sets, eliminating big evolving topics.

The miss probability is a minor issue for topic identification. Our goal is to *identify important topics* and to give a user the means to follow up on that topic. The low false alarm probability means that a story selected for follow-up *will* give good information on the topic. For the purpose of understanding general topics and trends in a corpus, it is more important to get all topics and a few good articles for each topic than to get all articles for a topic.

4.1 Comparison with TDT2 Systems

TopCat and the TDT2 participant systems are not directly comparable, as the TDT2 problem is online detection rather than TopCat's retrospective topic identification. The TDT2 systems are required to determine if a document fits in to an existing cluster or forms a new cluster after seeing 10 files beyond that document, where a file contains on average 36 stories (roughly corresponding to a news broadcast). Later work has shown that online detection does not make the TDT2 problem significantly harder [43]. The two TDT2 systems that were evaluated with both 10 and 100 file decision deferral verify this—the UIowa1 system showed a 1.5 percent improvement, but the UPenn1 system performed 49 percent *worse* with the longer deferral.

Table 6 shows the performance of TopCat and the eight TDT2 systems. TopCat figures are shown for named entities only, named entities with the addition of a set of keywords, and addition of the keywords expanded using WordNet (Section 3.1.2). TopCat is competitive at topic detection and provides a topic labeling ability not found in the other systems. This justifies our belief that the topics identified are comparable to what a person would expect.

4.2 Computational Requirements

Our implementation of TopCat is designed to test the concepts and was not optimized for performance. However, the speed of topic categorization is important. TopCat's use in the GeoNODE system [44] requires interactive clustering of user-defined subsets.

We would like to compare TopCat with document clustering systems. However, few of these systems report execution time figures. The web query clustering system Grouper [21] reports around 500 documents per second, but only for small numbers of documents (up to 800). How this would extend to large corpora that cannot fit in memory is unknown. The TDT topic detection reports do not include execution time. Informal discussions with TDT participants lead us to believe that TopCat is fast compared to the TDT systems.

We provide figures for the execution time of TopCat in clustering the entire TDT2 corpus. The TopCat prototype is designed for flexibility, not performance. All steps but named entity tagging and hypergraph clustering are implemented in SQL on a transaction-oriented commercial database. These times should be viewed as extreme upper bounds on the computational requirements. The times required on a Sun Ultra1/140 are:

1. Named Entity Tagging the entire 144MB TDT2 corpus took under 21 hours using Alembic. The machine received other use during this time, the

TABLE 6
TopCat and TDT2 System Results. TDT2 Systems Determined Topic after 10 Source Files, TopCat after Evaluating All Input

System		P_{Miss}	$P_{FalseAlarm}$	C_{Detect}
TopCat	*Named entities only*	0.1875	0.0021	0.0062
	Base keywords	0.1991	0.0016	0.0056
	Expanded key words	0.1962	0.0014	0.0053
BBN1		0.0941	0.0021	0.0040
CIDR1		0.3861	0.0018	0.0095
CMU1		0.3526	0.0004	0.0075
Dragon1		0.1638	0.0013	0.0045
IBM1		0.1965	0.0007	0.0046
UIowa1		0.6028	0.0009	0.0129
UMass1		0.0913	0.0022	0.0040
UPenn1		0.2997	0.0011	0.0070

normal rate is 128KB/minute. Alembic is a research prototype for applying machine learning techniques to identifying concepts in data. Existing commercial named entity tagging software is faster.

2. Coreference mapping required six hours 49 minutes. As others are working on better cross-document coreferencing, we have not tried to optimize this process.

3. Frequent itemset computation took 76 minutes. This can be improved using commercial data mining tools, as well as association rule algorithms specialized for text [45].

4. Hypergraph clustering of the TDT2 data took just under 5 minutes.

5. TFIDF-based cluster merging of clusters took 67 minutes. This was necessary to get good results on the TDT2 training data, but is not critical in practice.

Although the total process is computationally expensive, the most expensive parts are data preparation: named entity tagging and cross-document coreference computation. These are only done once per document and, in many systems (including GeoNODE), are done anyway for Information Retrieval and other purposes. The actual topic identification process is run more frequently: It is often interesting to manually define a subset of the corpus (e.g., a specific range of dates) and identify topics within that subset or to identify new topics and changes to existing topics as new articles are loaded. The most expensive part of the topic identification, computing frequent itemsets, can be significantly improved by raising the support threshold. If the goal is to identify only the 5-10 most important topics in a corpus, this is effective.

The current proof of concept implementation has proven adequate for real-world use in GeoNODE. Loading and tagging data is done as a background process. Topic identification on the entire corpus is done as a batch process and has been applied to over 300,000 documents. GeoNODE also uses TopCat to identify topics in a small subset (e.g., several hundred documents in a large topic or the results of a user query) on demand. While not truly interactive, it is "asynchronous interactive"—on the order of a minute, acceptable if a user can perform other tasks while waiting for topic identification results. A performance-oriented implementation of the frequent itemset generation and TFIDF-mapping stages (e.g., using commercially available

TABLE 7
Top Three Topics for January through June 1998

Topic 1		Topic 2		Topic 3	
LOCATION	Baghdad	LOCATION	Alaska	LOCATION	Albania
LOCATION	Britain	LOCATION	Anchorage	LOCATION	Macedonia
LOCATION	China	LOCATION	Caribbean	LOCATION	Belgrade
LOCATION	Iraq	LOCATION	Great Lakes	LOCATION	Bosnia
ORG.	Security Council	LOCATION	Gulf Coast	LOCATION	Pristina
ORG.	United Nations	LOCATION	Hawaii	LOCATION	Yugoslavia
PERSON	Kofi Annan	LOCATION	New England	LOCATION	Serbia
PERSON	Saddam Hussein	LOCATION	Northeast	PERSON	Slobodan Milosevic
PERSON	Richard Butler	LOCATION	Northwest	PERSON	Ibrahim Rugova
PERSON	Bill Richardson	LOCATION	Ohio Valley	ORG.	Nato
LOCATION	Russia	LOCATION	Pacific Northwest	ORG.	Kosovo Liberation Army
LOCATION	Kuwait	LOCATION	Plains		
LOCATION	France	LOCATION	Southeast		
ORG.	U.N.	LOCATION	West		
		PERSON	Byron Miranda		
		PERSON	Karen Mcginnis		
		PERSON	Meteorologist Dave Hennen		
		PERSON	Valerie Voss		

tools) would make such small-scale topic identification truly interactive.

5　CONCLUSIONS AND FUTURE WORK

We find the identified topics reasonable not only in terms of the TDT2-defined accuracy, but also understandable identifiers for the topic. For example, the most important three topics (based on the support of the frequent itemsets used in generating the topics) are shown in Table 7. The first (Iraqi arms inspections) is recognizable and gives information on the key players, although knowing that Richard Butler was head of the arms inspection team, Bill Richardson was the US Ambassador to the UN, and Saddam Hussein was the leader of Iraq may require viewing the documents; this shows the need to access documents based on the topic identifier. The third is also reasonably understandable: Events in and around Yugoslavia (note that this is a year before the NATO attacks on Serbia). The second topic is an amusing demonstration of the first half of the adage "Everybody talks about the weather, but nobody does anything about it." (Most television broadcasts included a weather segment.)

TopCat has since been applied to a variety of other corpuses as part of MITRE's GeoNODE project [44]. This includes non-English sources, Web-harvested data, broadcast news, newsgroups, and e-mail digests. The scope of the data has ranged from general (e.g., CNN broadcasts) to highly specialized (e.g., ProMed medical abstracts). The results are encouraging. While named entity extraction is sensitive to the type of corpus, TopCat is relatively insensitive to errors in named entity tagging. More critical to TopCat is the segmentation of stories—if many documents contains multiple unrelated stories, the TopCat results are unreliable. While segmentation of broadcast news has received considerable interest [46], [47], segmentation of other types of data (e.g., web pages, text) may also be a useful research topic. In spite of these difficulties, TopCat has proven useful in practice—GeoNODE has been (subjectively) evaluated and judged useful in real-world analytical environments [48].

Some of the components of TopCat have proven useful in ways beyond the original goals. The relationships described in Section 3.5 were developed to further coalesce the generated topics. We have also used them to construct hierarchies. Although their have been efforts to *classify* documents into hierarchies [49], construction of hierarchies has been a manual process.

Fig. 7 shows display of Parent/Child relationships from the GeoNODE project. This is taken from a collection of broadcast news, covering a longer period than the TDT data. Moving the mouse over a node shows the mnemonic for that topic, allowing a user to browse the relationships. The node size reflects the number of documents in the topic.

We have also tried another form of hierarchical clustering using TopCat. Given a large topic, we run TopCat against only documents in that topic. The high support threshold ignores the named entities that define the topic—the resulting topic identifiers are somewhat obscure as they are missing the most important named entities in the topics. However, within the context of a hierarchy, they are understandable and provide a useful drill-down capability.

The clustering methods of TopCat are not limited to topics in text; any market basket style problem is amenable to the same approach. For example, we could use the hypergraph clustering and relationship clustering on mail-order purchase data. This extends association rules to higher-level *related purchase* groups. Association rules provide a few highly specific *actionable items*, but are not as useful for high-level understanding of general patterns. The methods presented here can be used to give an overview of patterns and trends of related purchases, to use (for example) in assembling a targeted specialty catalog.

5.1　Future Work

One key problem we face is the continuity of topics over time. This raises two issues:

- Performance: Can we incrementally update the topics without looking at all the old data? The data mining community is addressing this for association

rules (for two examples, see [50] and [51]); this should apply directly to TopCat.

- New knowledge: How do we alert the user when something interesting has changed, either new topics or new information added to a topic?

We find the latter issue to be the greater challenge. For frequent itemsets, we can track when a new document results in a new (or modified) itemset. However, carrying this through the hypergraph partitioning and clustering is a difficult problem.

Another issue is using additional types of information. For example, the Alembic project is working on extracting events. How to best use this information is an open question. Grouping events into types (as we tried with keywords) may or may not be appropriate.

We have mapped documents into the market basket model using named entities. However, named entity processing really gives us a *typed* market basket (e.g., LOCATION or PERSON as types). We have used types only to distinguish between different entities with the same name (e.g., Clifton the person versus Clifton the city.) There may be additional ways to utilize this information. Another possibility is to use other generalizations (e.g., a geographic thesaurus equating Prague and Brno with the Czech Republic) in the mining process [52]. Further work on expanded models for data mining would have significant benefit for data mining of text.

ACKNOWLEDGMENTS

REFERENCES

[1] Y. Kodratoff, *Proc. European Conf. Machine Learning Workshop Text Mining,* Apr. 1998.

[2] R. Feldman and H. Hirsh, *Proc. IJCAI '99 Workshop Text Mining,* Aug. 1999.

[3] D. Mladenic and M. Grobelnik, *Proc. ICML-99 Workshop Machine Learning in Text Data Analysis,* June 1999.

[4] R. Feldman and H. Hirsh, "Exploiting Background Information in Knowledge Discovery from Text," *J. Intelligent Information Systems,* vol. 9, no. 1, pp. 83-97, July 1998.

[5] L. Singh, P. Scheuermann, and B. Chen, "Generating Association Rules from Semi-Structured Documents Using an Extended Concept Hierarchy," *Proc. Sixth Int'l Conf. Information and Knowledge Management,* Nov. 1997.

[6] H. Ahonen, O. Heinonen, M. Klemettinen, and I. Verkamo, "Mining in the Phrasal Frontier," *Proc. First European Symp. Principles of Data Mining and Knowledge Discovery (PKDD'97),* June 1997.

[7] B. Lent, R. Agrawal, and R. Srikant, "Discovering Trends in Text Databases," *Proc. Third Int'l Conf. Knowledge Discovery and Data Mining,* pp. 227-230, Aug. 1997.

[8] O. Zamir, O. Etzioni, O. Madan, and R.M. Karp, "Fast and Intuitive Clustering of Web Documents," *Proc. Third Int'l Conf. Knowledge Discovery and Data Mining,* pp. 287-290, Aug. 1997.

[9] E.-H. S. Han, G. Karypis, and V. Kumar, "Clustering Based on Association Rule Hypergraphs," *Proc. SIGMOD'97 Workshop Research Issues in Data Mining and Knowledge Discovery,* 1997.

[10] "1998 Topic Detection and Tracking Project (TDT-2)," July 1998, http://www.nist.gov/speech/tests/tdt/tdt98/.

[11] R. Hyland, C. Clifton, and R. Holland, "GeoNODE: Visualizing News in Geospatial Context," *Proc. Federal Data Mining Symp. and Exposition '99,* Mar. 1999.

[12] D. Lewis, W.B. Croft, and N. Bhandaru, "Language-Oriented Information Retrieval," *Int'l J. Intelligent Systems,* vol. 4, no. 3, pp. 285-318, 1989.

[13] M.L. Mauldin, "Retrieval Performance in FERRET: A Conceptual Information Retrieval System," *Proc. 14th Ann. Int'l ACM/SIGIR Conf. Research and Development in Information Retrieval (SIGIR '91),* pp. 347-355, Oct. 1991.

[14] E. Riloff and W. Lehnert, "Information Extraction as a Basis for High-Precision Text Classification," *ACM Trans. Information Systems,* vol. 12, no. 3, pp. 296-333, 1994.

[15] D.D. Lewis and K.S. Jones, "Natural Language Processing for Information Retrieval," *Comm. ACM,* vol. 39, no. 1, pp. 92-100, 1996.

[16] K. Kageura and B. Umino, "Methods of Automatic Term Recognition: A Review," *Terminology,* vol. 3, no. 2, 1996.

[17] G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval.* New York: McGraw-Hill, 1983.

[18] B. Hetzler, W.M. Harris, S. Havre, and P. Whitney, "Visualizing the Full Spectrum of Document Relationships," *Structures and Relations in Knowledge Organization: Proc. Fifth Int'l ISKO Conf.,* pp. 168-175, 1998, http://multimedia.pnl.gov:2080/infoviz/spire/spire.html.

[19] "Northern Light Search Help—Customer Search Folders," Dec. 2001, http://www.northernlight.com/docs/search_help_folders.html.

[20] Y. Wang and M. Kitsuregawa, "Link Based Clustering of Web Search Results," *Second Int'l Conf. Advances in Web-Age Information Management (WAIM 2001),* pp. 225-236, July 2001.

[21] O. Zamir and O. Etzioni, "Grouper: A Dynamic Clustering Interface to Web Search Results," *Proc. Eighth Int'l World Wide Web Conf.,* May 1999, http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html.

[22] D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, and M. Vilain, "Mixed Initiative Development of Language Processing Systems," *Proc. Fifth Conf. Applied Natural Language Processing,* Mar. 1997.

[23] Y. Yang and J.P. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," *Proc. 14th Int'l Conf. Machine Learning (ICML '97),* July 1997, http://www.cs.cmu.edu/yiming/papers.yy/ml97.ps.

[24] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Proc. European Conf. Machine Learning,* pp. 137-142, Apr. 1998.

[25] M. Porter, "An Algorithm for Suffix Stripping," *Automated Library and Information Systems,* vol. 14, no. 3, pp. 130-137, 1980.

[26] R. Cooley, "Classification of News Stories Using Support Vector Machines," *IJCAI '99 Workshop Text Mining,* Aug. 1999.

[27] G.A. Miller, C. Fellbaum, J. Kegl, and K.J. Miller, "Introduction to Wordnet: An On-Line Lexical Database," *Int'l J. Lexicography,* vol. 3, no. 4, pp. 235-244, 1990, ftp://ftp.cogsci.princeton.edu/pub/wordnet/5papers.ps.

[28] D. Harman, "Overview of the First Text REtrieval Conference (TREC-1)," *Proc. First Text REtrieval Conf. (TREC-1),* no. SN003-003-03614-5, Nat'l Inst. of Standards and Technology. Gaithersburg, Md.: Government Printing Office, pp. 1-20, Nov. 1992, http://trec.nist.gov/pubs/trec7/t7_proceedings.html.

[29] G. Salton, J. Allan, and C. Buckley, "Automatic Structuring and Retrieval of Large Text Files," *Comm. ACM,* vol. 37, no. 2, pp. 97-108, Feb. 1994, http://www.acm.org/pubs/citations/journals/cacm/1994-37-2/p97-salton/.

[30] K.W. Church and P. Hanks, "Word Association Norms, Mutual Information and Lexicography," *Computational Linguistics,* vol. 16, no. 1, pp. 22-29, 1991, http://www.research.att.com/kwc/published_1989_CL.ps.

[31] R. Feldman, Y. Aumann, A. Amir, A. Zilberstein, and W. Kloesgen, "Maximal Association Rules: A New Tool for Mining for Keyword Co-Occurrences in Document Collections," *Proc. Third Int'l Conf. Knowledge Discovery and Data Mining,* pp. 167-170, Aug. 1997.

[32] C. Silverstein, S. Brin, and R. Motwani, "Beyond Market Baskets: Generalizing Association Rules to Dependence Rules," *Data Mining and Knowledge Discovery,* vol. 2, no. 1, pp. 39-68, Jan. 1998.

[33] R. Agrawal, T. Imielinski, A.N. Swami, "Mining Association Rules between Sets of Items in Large Databases," *Proc. 1993 ACM SIGMOD Int'l Conf. Management of Data,* pp. 207-216, May 1993, http://www.almaden.ibm.com/cs/people/ragrawal/papers/sigmod93.ps.

[34] D. Tsur, J.D. Ullman, S. Abiteboul, C. Clifton, R. Motwani, S. Nestorov, and A. Rosenthal, "Query Flocks: A Generalization of Association Rule Mining," *Proc. 1998 ACM SIGMOD Conf. Management of Data,* pp. 1-12, June 1998.

[35] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," *Proc. 20th Int'l Conf. Very Large Data Bases,* Sept. 1994, http://www.vldb.org/dblp/db/conf/vldb/vldb94-487.html.

[36] R. Fano, *Transmission of Information.* Cambridge, Mass.: MIT Press, 1961.

[37] K. Wang, C. Xu, and B. Liu, "Clustering Transactions Using Large Items," *Proc. Eighth Int'l Conf. Information Knowledge Management,* pp. 483-490, Nov. 1999.

[38] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekar, "Multilevel Hypergraph Partitioning: Applications in VLSI Domain," *Proc. ACM/IEEE Design Automation Conf.,* 1997.

[39] "Topic Detection and Tracking: TDT Phase 2," July 2000, http://morph.ldc.upenn.edu/Projects/TDT2/.

[40] D.D. Lewis, "Evaluating Text Categorization," *Proc. Speech and Natural Language Workshop, Defense Advanced Research Projects Agency,* pp. 312-318, Feb. 1991.

[41] "The Topic Detection and Tracking Phase 2 (TDT2) Evaluation Plan," Nov. 1999, http://www.nist.gov/speech/tdt98/doc/tdt2.eval.plan.98.v3.7.pdf.

[42] J.M. Shultz and M. Liberman, "Topic Detection and Tracking Using IDF-Weighted Cosine Coefficient," *Proc. 1999 DARPA Broadcast News Workshop,* Feb. 1999. http://www.nist.gov/speech/publications/darpa99/html/abstract.htm#tdt3-10.

[43] V. Hatzivassiloglou, L. Gravano, and A. Maganti, "An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering," *Proc. 23rd Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* July 2000, http://www.cs.columbia.edu/gravano/Papers/2000/sigir00.pdf.

[44] C. Clifton, J. Griffith, and R. Holland, "GeoNODE: An End-to-End System from Research Components," *Demonstration Section 17th Int'l Conf. Data Eng.,* Apr. 1991.

[45] J.D. Holt and S.M. Chung, "Efficient Mining of Association Rules in Text Databases," *Proc. Eighth Int'l Conf. Information Knowledge Management,* pp. 234-242, Nov. 1999.

[46] "Topic Detection and Tracking Project (TDT),"http://www.nist.gov/speech/tests/tdt/index.htm, Sept. 2000.

[47] S. Boykin and A. Merlino, "Machine Learning of Event Segmentation for News on Demand," *Comm. ACM,* vol. 43, no. 2, pp. 35-41, Feb. 2000.

[48] L. Phillips, "Soft Copy Search and GeoNode," *Proc. Geospatial Intelligence Conf.—Uniting America's Defense, Assoc. of Old Crows,* Nov. 2002, http://www.crows.org/events_conf02geospatial.htm.

[49] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan, "Scalable Feature Selection, Classification and Signature Generation for Organizing Large Text Databases into Hierarchical Topic Taxonomies," *VLDB J.,* vol. 7, no. 3, pp. 163-178, Aug. 1998, http://www.almaden.ibm.com/cs/k53/irpapers/VLDB54_3.PDF.

[50] S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka, "An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases," *Proc. Third Int'l Conf. Knowledge Discovery and Data Mining,* pp. 263-266, Aug. 1997.

[51] R. Feldman, Y. Aumann, A. Amir, and H. Mannila, "Efficient Algorithms for Discovering Frequent Sets in Incremental Databases," *Proc. Workshop Research Issues on Data Mining and Knowledge Discovery (DMKD '97),* May 1997.

[52] R. Srikant and R. Agrawal, "Mining Generalized Association Rules," *Proc. 21st Int'l Conf. Very Large Databases,* Sept. 1995, http://www.almaden.ibm.com/cs/people/ragrawal/pubs.html#associations.

Chris Clifton received the PhD degree from Princeton University and the Bachelor's and Master's degrees from the Massachusetts Institute of Technology. He is an associate professor of computer science at Purdue University. While this work was being performed, he was a principal scientist in the Information Technology Center at the MITRE Corporation. Prior to joining MITRE in 1995, he was an assistant professor of computer science at Northwestern University. His research interests include data mining, database support for text, and database security. He is a senior member of the IEEE and a member of the IEEE Computer Society and the ACM.

Robert Cooley received the PhD degree in computer science from the University of Minnesota. His PhD dissertation represents some of the seminal work in Web mining and his papers are often quoted as the original references on topics such as Web usage mining and preprocessing for web mining. Dr. Cooley has been working on data mining of Web data since 1996. In addition to consulting for e-commerce companies and working as the director of data mining for an Internet startup, Dr. Cooley has published several papers on the topics of Web Usage Mining and Text Mining. Dr. Cooley is a member of the IEEE, IEEE Computer Society, and ACM.

Jason Rennie received the BS degree in computer science with a double major in mathematics from Carnegie Mellon University. He then began studying at the Massachusetts Institute of Technology, where he has attained an SM degree in electrical engineering and computer science and is currently pursuing the PhD degree in electrical engineering and computer science. He works at the Artificial Intelligence Laboratory with Tommi Jaakkola and is primarily interested in problems relating to text classification. His published works involve topics including data mining, Web spidering, and text classification. He is the author of "ifile," an open source tool for e-mail filtering.

▷ **For more information on this or any computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.