TopCat: Data Mining for Topic Identification in a Text Corpus*

Chris Clifton
The MITRE Corporation
clifton@mitre.org

Robert Cooley
University of Minnesota
cooley@cs.umn.edu

Jason Rennie
Massachusetts Institute of Technology
jrennie@ai.mit.edu

March 15, 2000

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 showing this technique is effective in identifying "topics" in collections of news articles.

1 Introduction

Data mining has emerged to address problems of understanding ever-growing volumes of information, finding patterns within the data that are used to develop useful knowledge. In particular, on-line textual data is growing rapidly, creating the need 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 semi-structured documents [5], and

*This work supported by the Community Management Staff's Massive Digital Data Systems Program. Work for this paper was performed while authors were at the MITRE Corporation.

- 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 help a human understand the collection, so a good solution must identify topics in some manner that is meaningful to a human. In addition, we want results that can be used for further exploration. This gives a requirement that we be able to identify source texts relevant to a given topic. This is related to document clustering [8], but the requirement for a topic *identifier* brings it closer to rule discovery mechanisms.

The way we apply data mining technology on this problem is to treat 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 occur together. Beginning with these frequent itemsets, we then cluster named entities based on their document inter-relations. This allows us to capture 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 is:

ORGANIZATION	Justice Department
PERSON	Janet Reno
ORGANIZATION	Microsoft

This is recognizable as the U.S. antitrust case against Microsoft. Although not as readable or informative as a narrative description of the topic, it is a compact, human-understandable representation. It also meets our "find the original documents" criteria, as the topic can 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. There are a few key developments that have wider application:

- The frequent itemset filtering criteria (Section 3.2.1).

- The hypergraph-based clustering mechanism, a more generalizable development of that 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 market basket style data mining problem.

We will next give some background on where this problem originated. In Section 3 we give details on the TopCat process. The process will be described start to finish, from issues such as data preparation and cleansing, to the mapping from topics back to documents. 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.

2 Problem Statement

The TopCat project started with a specific user need. A project called GeoNODE concerns itself with organizing news in a geographic fashion. Given a sequence of news stories, GeoNODE can produce a world map highlighting the locations that the stories discuss. In order to do this reliably GeoNODE must identify *ongoing* topics, since not every story on a topic explicitly mentions the geographical area in question. This directs us to exactly the problem at hand: providing some sort of human-understandable organization to a set of documents. Data mining experiments that we have conducted showed that using association rules can produce concepts that are identifiable as major news topics. This led us to develop a topic identification mechanism based on data mining techniques.

There are related topic-based problems being addressed. The Topic Detection and Tracking (TDT) project [10] looks at two specific problems:

Topic Tracking: Mapping incoming documents to a predefined set of topics, using a *training set* of documents already classified into topics.

Topic Detection: Recognizing 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, with the following exceptions:

- 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 have a requirement to identify each new document/topic as it arrives.

Even though our goals are slightly different, the test corpus developed for the TDT project (a collection of news articles manually classified into topics) provides a basis for us to evaluate

Figure 1: GeoNODE screen shot showing identified topics at lower right.

our work. A full description of the corpus can be found in [10]. For this evaluation, we use the topic detection criteria developed for TDT2 (described in Section 4). This requires that we go beyond identifying topics, and also match documents to a topic.

For purposes of using the TDT evaluation criteria, we define the problem as follows:

Definitions:

$Document : word+$

$TopicID : word+$

Data Source:

$Corpus : \{Document\}$

Goal: Produce the following functions

$TopicList(Corpus) : \{TopicID\}$

$Topicmatch(TopicList(Corpus), Document \in Corpus) : Topicid \subset TopicList(Corpus)$

In Section 4, we discuss how TopCat performs against the TDT criteria.

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

Both the use of natural language processing and term clustering have been studied extensively in the Information Retrieval (IR) domain, usually with the goal of increasing precision and recall [11, 12, 13]. Natural language processing has been used to automatically generate concept thesauri, generate document summaries, handle natural language queries, and perform feature space reduction for vector space models, as discussed in [14]. Term clustering has also been used for automatic thesaurus generation, as well as document clustering [15]. However, these techniques have had little use in attempts to understand a *collection*, as opposed to individual documents.

Referee 1: On related work, pg. 4, there is a brief discussion of how NLP and term clustering have been used in the past, and then a comment that the techniques have had little use in attempts to understand a collections... Exactly what has been done, and what has not. Who's work is closest to this and how to they relate to your fundamental research goals? It would also be helpful to have a very clear statement of the research contributions of this work; this would help delineate the related work.

3 Process

TopCat follows a multi-stage process, first identifying key concepts within a document, then grouping these to find topics, and finally mapping the topics back to documents and using the mapping to find higher-level groupings. Figure 2 gives an overview of this process. We identify key concepts within a document by using natural language techniques to extract named people, places, and organizations. This gives us a structure that can be mapped into a *market basket* style mining problem.[1] We then generate *frequent itemsets*, or groups of named entities that commonly appear together. Further clustering is done using a hypergraph splitting technique to identify groups of frequent itemsets that contain considerable overlap, even though not all of the items may appear together often enough to qualify as a frequent itemset.

The generated topics, a set of named entities, can be used as a query to find documents related to the topic (Section 3.4). Using this, we can identify topics that frequently occur in the same document to perform a further clustering step (identifying not only topics, but also topic/subtopic relationships).

We will use the following cluster, capturing professional tennis stories, as an example throughout this section.

PERSON	Andre Agassi
PERSON	Martina Hingis
PERSON	Mary Pierce
PERSON	Pete Sampras
PERSON	Venus Williams
PERSON	Serena
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. It comes from merging two subsidiary clusters (described in Section 3.5), formed from clustering seven frequent itemsets (Section 3.3).

3.1 Data Preparation

TopCat starts by identifying *named entities* in each article (using the Alembic[16] system). Alembic identifies people, locations, and organizations mentioned in the text based on linguistic cues, and identifies the type of entity (person, location, or organization) as well as

[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.

Figure 2: TopCat Process

the name. [2] This serves several purposes. First, it shrinks the data set for further processing. It also gives *structure* to the data, allowing us to treat documents as a set of typed and named entities. This gives us a natural database schema for documents that maps into the traditional market basket data mining problem. Third, and perhaps most important, it means that *from the start* we are working with data that is rich in meaning, improving our chances of getting human understandable results. We eliminate frequently occurring terms (those occurring in over 10% of the articles, such as United States), as these are used across too many topics to be useful in discriminating between topics.

Note that the use of named entities, as opposed to full text, is debatable. It has been shown that careful feature selection can slightly improve results in text categorization, while poor feature selection can have a large negative impact [17]. This leaves the question, are named entities a *good* form of feature selection?

We tested this on our dataset using Support Vector Machines as classifiers [18]. Using the TDT2 training/development sets as our training and test sets (stemmed using the Porter stemming algorithm [19], and filtered for a list of common stopwords), we obtained a precision of 95% for full text categorization, versus 82% for named entity based categorization (the recall was nearly identical, at 87% and 86% respectively): Full text was better than named entities. Details of this test are given in [20].

However, for the problem of topic *identification*, the use of full text is not nearly as clear cut. We tested TopCat with full text, and found two problems. The first was with scalability (the stemmed/filtered full text corpus contained almost 5 million unique word-document pairs vs. 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% support it took nine hours on full text, and found only a single two-itemset.) We tried a smaller test set (one week of data), and the TopCat process took approximately one hour at 2% support. Using named entities from the same data took only two minutes at 0.5% support.

More critical is the difference in the *quality* of the results. Using 2% support and full-text generated 91 topics, many of which were nonsensical such as (tip, true) and (chat, signal, insid) or non-topic 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 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 feel the named entity version is just as recognizable, and contains more *useful*

[2] Although not tested specifically on the TDT2 corpus, Alembic and other top Named Entity tagging systems typically achieve 90-95% precision and recall.

Table 1: Asian Economic Crisis Topic: Full Text vs. 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

information.

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 combine such references before we proceed.

There are two issues involved:

1. How do we identify multiple references to the same entity within a document (as shown above); 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 between items where the *predicted* item is a substring of the predictor. This works well for person names, where the short name is uncommon, but is less effective with organization 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 (solving problem 1 above). Problem 2 is more difficult. Some choices clearly will not work (Marcelo Rios is referred to as Marcelo only once in our corpus). However, choosing the most common version doesn't work either (he is referred to as Marcelo Rios 82 times, and Rios 264; but there are 73 references to Rios that refer to someone else).

Therefore we 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 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 (in fact, the difference between the above two approaches with respect to the TDT2 evaluation criteria is small).

Although the natural language technique is our primary approach, we also use the association rule based approach with a minimum support of 0.05% and a minimum confidence of 50%. This only results in six additional translations, but as they are relatively frequent it does affect results. Most are straightforward (e.g., sports team full names versus short names, such as New York Rangers vs. Rangers); these are frequently abbreviated in short articles, and thus are missed by the natural-language "single document" technique. The only questionable one was a translation of Korea to South Korea; a sample of the documents affected showed this to be appropriate.

3.1.2 Data Cleansing

In addition to identifying the named entities within each document, several data cleaning algorithms were applied prior to the knowledge discovery phase in order to increase the quality of the results. The data cleaning techniques used by the TopCat system, shown in Figure 3, can be broadly classified into two categories, *generic* and *domain specific.* The generic data cleaning techniques are mostly taken from the Information Retrieval domain, and include case normalization and a stop algorithm. Since words traditionally found in a stop list, such as articles of speech and prepositions, have already been removed due to named entity identification, TopCat uses a simple stop algorithm. The stop algorithm takes a user supplied upper bound for the percentage of documents that contain any given term (referred to as *document frequency*), and removes all terms that have a document frequency greater than the upper bound. Frequently occurring terms, such as United States, are generally used across too many subjects to be useful in discriminating between topics; removing these improved our results. The notion of the document frequency being inversely proportional to the usefulness or information gain of the term is the basis for Salton's popular TFIDF (term frequency/inverse document frequency) term weighting scheme [21] that is used in many Information Retrieval applications.

Figure 3: TopCat Data Cleaning

The *domain specific* data cleaning steps used by TopCat for the TDT test corpus are removal of duplicate stories (an artifact of pulling stories off of a “newswire”, where errors cause the entire story to be retransmitted) and removal of what will be referred to as *composite* stories. A *composite* story is a multi-topic story that contains brief descriptions or recaps of stories reported on elsewhere. In the print media domain, 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 across 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 multi-topic 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 re-occurring 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 our case, a frequent itemset is a group of named entities that occur together in multiple articles. What this really gives us is correlated items, rather than any notion of a topic. However, we found that correlated named entities frequently occurred within a recognizable topic.

Discovery of frequent itemsets is a well-understood data mining problem, arising in the *market basket* association rule problem [22]. A document can be viewed as a market basket of named entities; existing research in this area applies directly to our problem. We perform the search directly in a relational database using *query flocks*[23] technology, allowing us to incorporate the filtering criteria described below into the search while relying on the database query processor for many algorithmic issues. One problem with frequent itemsets is that the items must co-occur *frequently*, causing us to ignore topics that occur in only a few articles. To deal with this, we use a low support threshold of 0.05% (25 occurrences in the TDT corpus). 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 too many frequent itemsets (6028 2-itemsets in the TDT corpus). We need additional filtering criteria to get just the "important" itemsets.[3]

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; and

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

We have already discussed problems with support. 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.

We use *interest*[25], a measure of correlation strength (specifically, the ratio of the probability of a frequent itemset occurring in a document to the multiple of the independent probabilities of occurrence of the individual items) as an additional filter. This emphasizes relatively rare items that generally occur together, and de-emphasizes common items. We select all frequent itemsets where either the support or interest are at least one standard deviation above the average, or where both support and interest are above average (note that this is computed independently for 2-itemsets, 3-itemsets, etc.) For 2-itemsets, this brings us from 6028 to 1033. This is still dependent on the choice of a minimum support; computing this *efficiently* without a fixed minimum support is an interesting problem.

We also use interest to choose between "contained" and "containing" itemsets (i.e., any 3-itemset contains three 2-itemsets with the required support.) We don't need to keep the 2-itemsets independently; however, a strong 2-itemset is better than a weak 3-itemset. An $n-1$-itemset is used only if it has greater interest than the corresponding n-itemset, and an n-itemset is used only if it has greater interest than at least one of its contained $n-1$-itemsets. This brings us to 416 (instead of 1033) 2-itemsets.

The difficulty with using frequent itemsets for topic identification is that they tend to be over-specific. For example, the "tennis player" frequent itemsets consist of the following:

[3] The problems with traditional data mining measures for use with text corpuses have been noted elsewhere as well, see [24] for another approach.

Type1	*Value1*	*Type2*	*Value2*	*Support*	*Interest*
PERSON	Andre Agassi	PERSON	Marcelo Rios	.00063	261
PERSON	Andre Agassi	PERSON	Pete Sampras	.00100	190
PERSON	Anna Kournikova	PERSON	Martina Hingis	.00070	283
PERSON	Marcelo Rios	PERSON	Pete Sampras	.00076	265
PERSON	Martina Hingis	PERSON	Mary Pierce	.00057	227
PERSON	Martina Hingis	PERSON	Serena	.00054	228
PERSON	Martina Hingis	PERSON	Venus Williams	.00063	183

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 interest; these are locations of matches and home countries of players (interesting, perhaps, but again relevant to specific matches rather than "championship tennis" as a whole.)

3.3 Clustering

We experimented with different frequent itemset filtering techniques, but were always faced with an unacceptable tradeoff between the number of itemsets and our ability to capture a reasonable breadth of topics. Further investigation showed that some named entities we should group 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 in addition to the discovery of frequent itemsets. 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 takes advantage of the computational effort already performed by the generation of 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 for a hyperedge. Clusters can be quickly found by using a hypergraph partitioning algorithm such as hMETIS [26].

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 do not have any directionality and therefore do not need to be combined prior to being used in a hypergraph. The interest of each itemset was used for the weight of each edge. Since interest tends to increase dramatically as the number of items in a frequent itemset increases, the log of the interest was used in the clustering algorithm to prevent the larger itemsets from completely dominating the process.

Upon investigation, we found that the stopping criteria presented in [9] only works for domains that form 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* if 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.) It 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 looks 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 2 vertices. (A cluster of one item is not particularly 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 user-defined *cutoff* parameter is used which 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 of m edges e, and c 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 remains in a partition if 2 or more vertices from the original edge are in the partition. For example, a cut-weight ratio of 0.5 means that the weight of the cut edges is half of 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.

Figure 4: Hypergraph of Tennis Player Frequent Itemsets

- Once the stopping criteria has been reached for all of the partitions of a hypergraph, vertices can be “added back in” to clusters depending on the user-defined *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%, and 3 of the original 4 vertices of an edge end up in the same cluster, the 4th 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).

For our domain, we found that the results were fairly insensitive to the cutoff criteria. Cut-weight ratios from 0.3 to 0.8 produced similar clusters, with the higher ratios partitioning the data into a few more clusters than the lower ratios. The hypergraphs that are created from the tennis player frequent itemsets are shown in Figure 4. Note that in this example, each hypergraph becomes a single cluster. Cuts *are* performed before the stopping criteria is reached. For example in the “men’s” cluster, the Agassi/Sampras and Agassi/Rios links are cut. However, they are added back in the final step. The end result is the clusters below.

Figure 5: Hypergraph of New York Yankees Baseball Frequent Itemsets

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

The TDT data produced one huge hypergraph containing half the clusters. Most of the rest are independent hypergraphs that become single clusters. Although the large hypergraph demonstrates the utility of this method, it is too large to use as an example. One that does not become a single cluster is shown in Figure 5. Here, the link between Joe Torre and George Steinbrenner (shown dashed) is cut. Even though this is not the weakest link, the attempt to balance the graphs causes this link to be cut, rather than producing a singleton set by cutting a weaker link. This is a sensible distinction. For those that don't follow U.S. baseball, George Steinbrenner is the owner of the New York Yankees, and Joe Torre is the manager. Darryl Strawberry and David Cone are 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 to a crumbling Yankee Stadium back in New York – thus the end result *does* reflect what is really happening.

3.4 Mapping to Documents

The preceding process gives us reasonable topics. However, to evaluate this with respect to the TDT2 instrumented corpus, we must map the identified topics back to a set of documents. We could trace back to the source data, using the fact that frequent itemsets can be tracked directly to a set of documents. However, this had two problems:

1. a document can be responsible for multiple frequent itemsets, 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 the TFIDF metric[4] to generate a distance measure between a document and a topic, then choose the closest topic for each document. This is a flexible measure; if desired, we can use cutoffs (a document isn't close to any topic), or allow multiple mappings.

Note that this is all done within the database; we never need to refer back to the full text.

3.5 Combining Clusters based on Document Mapping

Although the clustered topics appeared reasonable, we were over-segmenting with respect to the TDT "ground truth" criteria. 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
PERSON	Pete Sampras
PERSON	Venus Williams
PERSON	Serena
PERSON	Marcelo Rios
PERSON	Anna Kournikova

[4] The TFIDF weight between a document i and topic t is calculated as follows:[21]

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

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.

There are two types of merge. 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.)

The calculation of these values is actually a bit more complex, as it also takes into account negative relationships (two marriage topics are not only close to the same documents, but also far away from the same documents.)

3.5.1 Marriage Relationship Calculation

The marriage similarity between clusters a and b is defined as:

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

Based on experiments on the TDT2 training set, we chose a cutoff of 30 ($Marriage_{ab} \geq 30$) for merging clusters. This value can be adjusted depending on user requirements. Note that 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, since merging clusters is done by taking a union of the named entities in the two, and there is no requirement that the topic identifiers partition the set of entities, it is not a practical issue (we end up with two topics instead of the original three). We do merge into a single cluster where such transitivity exists.

3.5.2 Parent/Child Relationship Calculation

The parent child relationship is calculated as follows:

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

We calculate the parent/child relationship after the marriage clusters have been merged. In this case, we used a cutoff of 0.3 (this was a reasonably easy choice; the highest similarity was 0.4, and the closest to 0.3 were 0.27 and 0.35 – a natural break.) Merging the groups is again accomplished through a union of the named entities.

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

4 Experimental Results

The TDT2 evaluation criteria is based on the probability of failing to retrieve a document that belongs with the topic, and the probability of erroneously matching a document to the topic. These are combined to a single number C_{Det} as follows [27]:

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

where:

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

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

H is the set of stories associated with a TopCat-produced topic.

P_{topic} (the *a priori* probability of a story being on some given topic) = 0.02.

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

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

The mapping $H(R)$ between TopCat-identified topics and reference topics is defined to be the mapping that minimizes C_{Det} for that topic (as specified by the TDT2 evaluation process):

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

where

$$\begin{aligned} C_{Det}(R, H) &= C_{Miss} \cdot P_{Miss}(R, H) \cdot P_{topic} + C_{FalseAlarm} \cdot P_{FalseAlarm}(R, H) \cdot (1 - P_{topic}) \\ P_{Miss}(R, H) &= N_{Miss}(R, H)/|R| \\ P_{FalseAlarm}(R, H) &= N_{FalseAlarm}(R, H)/|S - R| \end{aligned}$$

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

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

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

S is the set of stories to be scored in the evaluation corpus being processed.

Using the TDT2 evaluation data (May and June 1998), the C_{Det} 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 5. This was comparable to the results from the TDT2 topic detection participants[28], which ranged from 0.0040 to 0.0129, although they are not directly comparable (as the TDT2 topic detection is on-line, rather than retrospective). Of note is the low false alarm probability we achieved (0.002); further improvement here would be difficult. The primary impediment to a better overall score is the miss probability of 0.17.

Although the use of keywords did provide some improvement in the scores on the evaluation set, it was not significant. Only two topics that mapped to evaluation topics contained keywords; one of these (mapping California primaries to the Unabomber case) was a mistake. The other added the terms earthquake and quake to the topic matching the "Afghan Earthquake" topic. However, some topics (corresponding to topics *not* part of the TDT2 list) did include interesting keywords, such as suit to the Microsoft anti-trust case.

The primary 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 matched 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). TopCat often identified separate topics, such as (for the conflict with Iraq) Madeleine Albright/Iraq/Middle East/State, in addition to the "best" topic (lowest C_{Det} score) shown at the top of Table 3. The TFIDF-based topic merging of Section 3.5 addressed this to some extent, substantially improving results in the training set. (Interestingly, the topic merging didn't have a significant effect on the evaluation set.) Although various TopCat parameters could be changed to merge these, many similar 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 miss probability is a minor issue for our problem. 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.

5 Keywords

Named entities can be very powerful identifiers for allowing a human to identify a topic or event. However, they are not all-encompassing. Named entities can capture questions like "Who?" and "Where?", but require that we use our background knowledge and inference capabilities to answer questions such as "What?" "When?" and "Why?" Other words in documents can answer these questions, but, as has been mentioned before, data mining techniques have difficulty dealing with the unrestricted set of information found in a large document collection. Named entities are one manageable subset. Another possibility is to have a human generate a small set of keywords that are related to the sorts of documents in our corpus. Since we value efficiency, we ask for a small set of keywords that can be quickly generated. An expansive keyword set might require significant human effort. This set of keywords provides us with another manageable subset of information. However, this keyword set is likely to be limited; even though the words may adequately describe the topic of interest, numerous other words are likely to be used to refer to the same idea. This leads us to the problem of automatically expanding the keyword list to better represent the full list of words that might be used to describe the topic.

WordNet [29] is a tool that can help us make use of these keywords without overburdening the data mining system or asking the human to provide an expansive list of keywords. WordNet is a semantic network developed by George Miller at Princeton University that has mapped out a significant portion of the English language in a hierarchical lexicon. It includes linkages that we are all familiar with, such as synonyms and antonyms, along with more powerful, yet lesser known relations, such as hypernyms and hyponyms. A hypernym is a word that is more general than another word. Similarly, a hyponym is a word that is more specific. For example, we can say that vehicle is a hypernym of automobile and that couch is a hyponym of furniture. By exploiting these relations, we can expand a set of keywords to include related words that were not part of the original keyword set. WordNet has been continually honed and improved during the 14 years of its existence and now includes relations for over 100,000 word forms. The hyper/hyponym noun hierarchy is the most developed portion of WordNet and will be the portion that we rely upon for our expansion rules.

Although no such collection could possibly be complete, WordNet holds an expansive amount of knowledge about the human language. Given a term such as politician, it can tell us that a politician is a type of leader and that Democrats and Republicans are types of politicians. It might be difficult to locate a document on U.S. politics using simply the query politician. Words such as Democrat, Republican and leader can allow us to match more related documents (increase recall) without much damaging the specificity (decrease precision) In the next few paragraphs, we describe how WordNet is used for the keyword expansion task and provide some experimental evidence that WordNet is a viable resource for expanding a set of keywords to better represent the corresponding topic.

While WordNet is a great repository of knowledge, it is not a panacea for the keyword expansion problem. The knowledge that it stores was constructed by humans without any underlying mathematical model. Thus, some of the symmetry that one might expect, such as equal semantic distance between links, do not hold. Since we know that WordNet is a valuable resource, we have spent time determining which of its aspects are most useful for the task of keyword expansion. As a result of this work, 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.

We find the basis for our first heuristic in the word sense disambiguation (WSD) literature. This literature is often closely tied to WordNet because it is the only large scale, well-known, publicly available lexicon that describes interconnections between individual senses of words. Many WSD papers, such as [30], [31] and [32] use WordNet and the Brown corpus [33] (the document base used to construct WordNet) as a foundation for their evaluations. One observation that has been made in the WSD literature is that words are used to mean their most common sense about 80% of the time. Since it would not be realistic to expect our document corpus to be sense disambiguated, we do not want to expand our keyword set with words that are likely to retrieve irrelevant documents. For example, WordNet gives the sixth sense of force as a synonym for violence. Since force is more commonly used to refer to a powerful effect or influence, we would not want to expand violence with a word such as force. Since we cannot expect that the search corpus could be fully disambiguated, this heuristic limits the degree to which we add misleading words to the keyword set.

The WordNet hyper/hyponym relations form a set of directed acyclic graphs (DAGs). In order to gain some understanding of the structure, we assign these designated root words a *depth* of 1. We then define the depth of any word to be one plus the depth of its shallowest hypernym. Using this definition, a word such as as clothing a depth of 5. After conducting informal keyword expansion experiments, we determined that larger semantic leaps are most commonly found near in shallower regions of each DAG. We performed hyper/hyponym expansions on the topic statement for 20 TREC queries and qualitatively evaluated each expansion word for relevance. As low as depth 4 we found fairly misleading hypernyms, such as clothing @ → covering. Hence, we decided to only allow hypernym expansion for words below depth 5 (hypernyms below depth 4). It should be noted that the WordNet hyper/hyponym hierarchy is quite expansive and runs many levels deep. Of the 35 hypernyms that we considered for inclusion in the expanded keyword set, 23 were at or below depth 5.

Table 2: Performance with Keywords

Data Used	*Story weighted*			*Topic weighted*		
	CDET	*Miss*	*False*	*CDET*	*Miss*	*False*
Named entities only	.0062	.19	.0025	.0089	.32	.0025
Base keywords	.0056	.20	.0016	.0088	.36	.0017
Expanded keywords	.0053	.20	.0014	.0076	.31	.0014

Just as hypernym relations can provide us with words that can misrepresent the given topic, so too can hyponym relations. To reduce the possibilities of adding such a word to our expanded keyword list, we restrict the set of hypernym relations that we are willing to consider. When a word has a small handful of hyponyms, it is likely that those words are closely related to the topic at hand. However, when the word has a large number of hyponyms, it is more likely that some of those words are misleading. Clothing has 29 hyponyms and is a perfect example of where this problem strikes. Many are words that would be useful for further expanding a topic related to clothing, but some may significantly alter the topic at hand. One such misleading hyponym of clothing is G-string. An altavista search on G-string leads us to lingerie, porn and music, hardly what one would think of given a topic of clothing. In order to restrict such occurrences, we decide to not expand the hyponym relations of words that have more than 15 hyponyms. This heuristic will help our expansion from blowing out of proportion.

These heuristics give us a set of rules that should give us a fairly robust. For example, given the keyword set president, U.S., WordNet 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, obviously a significant improvement in breadth without the sacrifice in precision that one might find in other keyword expansion techniques.

In order to evaluate this keyword expansion technique, we have run experiments on the TDT data according to standard evaluation metrics. Table 2 lists the results from such experiments using the named entities only, using the keywords and named entities (**base keywords**) and using the expanded keywords and named entities (**expanded keywords**). As one can see, using a short list of keywords and our heuristic keyword expansion technique both improve the overall **CDET** score. Most impressive are the "topic weighted" results, where the additional expansion words reduce the **CDET** score by 14%[5]. Hence keyword expansion does improve, indicating that keyword expansion can expand our topic coverage without drawing in irrelevant or tangential ideas.

[5] Lower **CDET** scores are better

Figure 6: Types of Relationships

6 Constructing Hierachies of Topics

The relationships described in Section 3.5 were developed to further coalesce the generated topics. However, a more important use is to construct hierarchies. Although work has been done in *classifying* documents into hierarchies [34], construction of the hierarchies has been a manual process.

These relationships capture two different types of overlap between topics. The *marriage* relationship occurs when there is a high degree of overlap between the documents. The *parent/child* relationship happens when one topic is a subset of another. A graphic description of the types of relationships is given in Figure 6.

We had 47 pairs with similarity greater than 30 for the marriage relationship in the TDT data. 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 showin in Table 3.

The two examples with highest similarity are:

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	

The Parent/Child relationship gave 16 pairs with a similarity greater than 0.3 in the TDT data. These are divided into 7 hierarchies. The highest similarity three groups are shown below (note that the India/Pakistan topic has two children):

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			

A display developed for the GeoNODE project capturing Parent/Child relationships is shown in Figure 7. 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 captures the number of documents associated with the topic.

7 Conclusions and Future Work

We find that the identified topics are not only reasonable in terms of the TDT2 defined accuracy, but also are understandable identifiers for the subject. For example, the most important three topics (based on the support of the frequent itemsets used in generating the topics) are apparent from Table 3. The first (Iraqi arms inspections) also gives information on who is involved (although knowing that Richard Butler was head of the arms inspection team, Bill Richardson is the U.S. Ambassador to the UN, and Saddam Hussein is the leader of Iraq may require looking at the documents; this shows the usefulness of being able to access documents based on the topic identifier.) The third is also reasonably understandable: Events in and around Yugoslavia. The second is an amusing proof of the first half of the adage “Everybody talks about the weather, but nobody does anything about it.”

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.

Figure 7: Display of Relationships found in Broadcast News

Table 3: Top 3 Topics for January through June 1998

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

7.1 Computation Requirements

Our implementation of TopCat is designed to test the concepts, and not performance. We have used research software designed for flexibility, not performance. In particular, all but the named entity tagging and hypergraph clustering are implemented in SQL and run on a transaction-oriented commercial database. Thus these times should be viewed as *extreme* upper bounds on the computational requirements. However, for those interested, we give some ideas of the times required (all times an a Sun Ultra1/140):

Named Entity Tagging Alembic tagged the entire 144MB TDT2 corpus in under 21 hours. However, the machine was heavily used during some of this time, so this number is high. A figure of 128KB/minute would be more appropriate. However, Alembic is a research tool for applying machine learning techniques to identifying concepts in data, and is not optimized for performance. Commercial named entity tagging software exists that would do better.

Coreference mapping The coreference mapping procedure described six hours 49 minutes. As this is not a central feature of this work, and others are working on better ways of doing cross-document coreferencing, we have not worried about the expense of this process.

Frequent itemset computation Computing frequent itemsets to 76 minutes. However, this could easily be improved using highly optimized commercial data mining tools (TDT2 has 1.5 million named entities and 65000 documents – an easy task for commercial "market basket association rule" tools).

Hypergraph clustering The hypergraph clustering step took just under 5 minutes on the TDT2 data.

TFIDF-based cluster merge The TFIDF-based merging of clusters took 67 minutes. Although we found this necessary to get reasonable results on the TDT2 training data, we find that the results obtained without this step *are* meaningful to humans, even if they are more specific than the human breakdowns of topics in the TDT2 testbed (this was validated on the TDT2 evaluation set.) This step would be primarily of interest for developing the topic hierarchies in Section 6.

Although the total process is computationally expensive, the most expensive parts are data preparation: Named entity tagging and cross-document coreference computation. These need only be done once per document. The actual topic identification process is done more frequently: it is often interesting to manually define a subset of the corpus (e.g., a specific range of dates), then identifying topics within that subset; or identifying new topics and changes to existing topics as new articles are loaded. In addition, 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. A practical application of TopCat would involve continuously loading/tagging data as a background process. Given this, topic identification – while not truly interactive – could easily be fast enough to be done "on demand" whenever a corpus subset of interest is identified.

7.2 Future Work

One key problem we face is the continuity of topics over time. There are two issues here:

- 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 [35] and [36]); this should apply directly to TopCat.

- New knowledge: How do we alert the user when something interesting has changed?

We find the latter issue to be the greater challenge. There are two types of changes: New topics, and new information added to a topic. For frequent itemsets, this is feasible. One approach would be to identify when a new frequent itemset is a result of documents that contributed to an old frequent itemset, with some new documents giving support to a change (new information added to the itemset), or a result of documents that did not previously support an itemset (a new itemset). However, carrying this through the hypergraph partitioning / clustering is a difficult problem.

Another issue is the type of information to use. We have shown that using all words is not appropriate, but extracting more information should help. As information extraction technology advances, we will be able to make use of information other than named entities and user-provided keywords. For example, the Alembic project is working on extracting events. How to best make use of this information is an open question. For example, grouping events into types (as we did with keywords) may or may not be appropriate.

We have talked about how we map documents into the market basket model using named entities. However, what the named entity processing really gives us is a *typed* market basket (e.g., LOCATION or PERSON as types.) We have made little use of the types, but their presence could be beneficial. Another possibility is to use generalizations (e.g., a geographic "thesaurus" equating Prague and Brno with the Czech Republic) in the mining process[37]. As the extracted information becomes richer, these issues will increase in importance. Further work on expanded models for data mining could have significant impact on data mining of text.

Acknowledgments

We would like to thank Marc Vilain, David Day, and other members of the MITRE Alembic team for their work in Information Extraction. We would also like to that Robert Hyland and others of the GeoNODE project for making use of TopCat, and providing the displays shown in Figures 1 and 7.

References

[1] Yves Kodratoff, Ed., *European Conference on Machine Learning Workshop on Text Mining*, Chemnitz, Germany, Apr. 1998.

[2] Ronen Feldman and Haym Hirsh, Eds., *IJCAI'99 Workshop on Text Mining*, Stockholm, Sweden, Aug. 2 1999.

[3] Dunja Mladenić and Marko Grobelnik, Eds., *ICML-99 Workshop on Machine Learning in Text Data Analysis*, Bled, Slovenia, June 30 1999.

[4] Ronen Feldman and Haym Hirsh, "Exploiting background information in knowledge discovery from text," *Journal of Intelligent Information Systems*, vol. 9, no. 1, pp. 83–97, July 1998.

[5] Lisa Singh, Peter Scheuermann, and Bin Chen, "Generating association rules from semi-structured documents using an extended concept hierarchy," in *Proceedings of the Sixth International Conference on Information and Knowledge Management*, Las Vegas, Nevada, Nov. 1997.

[6] Helena Ahonen, Oskari Heinonen, Mika Klemettinen, and Inkeri Verkamo, "Mining in the phrasal frontier," in *1st European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'97)*, Trondheim, Norway, June 25–27 1997.

[7] Brian Lent, Rakesh Agrawal, and Ramakrishnan Srikant, "Discovering trends in text databases," in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, Aug. 14–17 1997, pp. 227–230.

[8] Oren Zamir, Oren Etzioni, Omid Madan, and Richard M. Karp, "Fast and intuitive clustering of web documents," in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, Aug. 14–17 1997, pp. 287–290.

[9] Eui-Hong (Sam) Han, George Karypis, and Vipin Kumar, "Clustering based on association rule hypergraphs," in *Proceedings of the SIGMOD'97 Workshop on Research Issues in Data Mining and Knowledge Discovery.* 1997, ACM.

[10] "1998 topic detection and tracking project (TDT-2)," Nov. 1999, http://www.nist.gov/speech/tdt98/tdt98.htm.

[11] D. Lewis, W. B. Croft, and N. Bhandaru, "Language-oriented information retrieval," *International Journal of Intelligent Systems*, vol. 4, no. 3, pp. 285–318, 1989.

[12] Michael L. Mauldin, "Retrieval performance in FERRET: A conceptual information retrieval system," in *SIGIR'91. Proceedings of the fourteenth annual international ACM/SIGIR conference on Research and development in information retrieval*, Chicago, Illinois, Oct. 13-16 1991, pp. 347–355.

[13] Ellen Riloff and Wendy Lehnert, "Information extraction as a basis for high-precision text classification," *ACM Transactions on Information Systems*, vol. 12, no. 3, pp. 296–333, 1994.

[14] David D. Lewis and Karen Sparck Jones, "Natural language processing for information retrieval," *Communications of the ACM*, vol. 39, no. 1, pp. 92–100, 1996.

[15] Gerard Salton and Michael J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company, New York, 1983.

[16] David Day, John Aberdeen, Lynette Hirschman, Robyn Kozierok, Patricia Robinson, and Marc Vilain, "Mixed initiative development of language processing systems," in *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, D.C., Mar. 1997.

[17] Yiming Yang and Jan P. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, Jr. Doughals H. Fisher, Ed., Nashville, TN, July 8–12 1997.

[18] Thorsten Joachims, "Text categorization with support vector machines: Learning with man y relevant features," in *Proceedings of the European Conference on Machine Learning*, Chemnitz, Germany, Apr. 1998, pp. 137–142, Springer-Verlag.

[19] M. Porter, “An algorithm for suffix stripping,” *Automated Library and Information Systems*, vol. 14, no. 3, pp. 130–137, 1980.

[20] Robert Cooley, “Classification of news stories using support vector machines,” in *IJCAI’99 Workshop on Text Mining*, Stockholm, Sweden, Aug. 2 1999.

[21] Gerard Salton, James Allan, and Chris Buckley, “Automatic structuring and retrieval of large text files,” *Communications of the ACM*, vol. 37, no. 2, pp. 97–108, Feb. 1994.

[22] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami, “Mining association rules between sets of items in large databases,” in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Peter Buneman and Sushil Jajodia, Eds., Washington, D.C., May 26–28 1993, pp. 207–216.

[23] Dick Tsur, Jeffrey D. Ullman, Serge Abiteboul, Chris Clifton, Rajeev Motwani, Svetlozar Nestorov, and Arnon Rosenthal, “Query flocks: A generalization of association rule mining,” in *Proceedings of the 1998 ACM SIGMOD Conference on Management of Data*, Seattle, WA, June 2-4 1998, pp. 1–12.

[24] Ronen Feldman, Yonatan Aumann, Amihood Amir, Amir Zilberstein, and Wiolli Kloesgen, “Maximal association rules: a new tool for mining for keyword co-occurrences in document collections,” in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, Aug. 14–17 1997, pp. 167–170.

[25] Craig Silverstein, Sergey Brin, and Rajeev Motwani, “Beyond market baskets: Generalizing association rules to dependence rules,” *Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 39–68, Jan. 1998.

[26] George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekar, “Multilevel hypergraph partitioning: Applications in VLSI domain,” in *Proceedings of the ACM/IEEE Design Automation Conference*, 1997.

[27] “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.

[28] “The topic detection and tracking phase 2 (TDT2) evaluation,” ftp://jaguar.ncsl.nist.gov/tdt98/tdt2_dec98_official_results_19990204/index.htm.

[29] George A. Miller, Christiane Fellbaum, Judy Kegl, and Katherine J. Miller, “Introduction to wordnet: an on-line lexical database,” *International Journal of Lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[30] Rada Mihalcea and Dan Molovan, “A method for word sense disambiguation of unrestricted text,” in *Proceedings of ACL ’99*, 1999.

[31] Xiaobin Li, Stan Szpakowicz, and Stan Matwin, “A WordNet-based algorithm for word sense disambiguation,” in *Proceedings of the Twelth International Joint Conference on Artifi cial Intelligence (IJCAI-95)*, 1995.

[32] Eneko Agirre and German Rigau, “Word sense disambiguation using conceptual density,” in *COLING*, 1996.

[33] S. Francis and H. Kucera, *Computing Analysis of Present-day American English*, Brown University Press, Providence, RI, 1967.

[34] Soumen Chakrabarti, Byron Dom, Rakesh Agrawal, and Prabhakar Raghavan, “Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies,” *VLDB Journal*, vol. 7, no. 3, pp. 163–178, Aug. 1998.

[35] Shiby Thomas, Sreenath Bodagala, Khaled Alsabti, and Sanjay Ranka, “An efficient algorithm for the incremental updation of association rules in large databases,” in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, Aug. 14–17 1997, pp. 263–266.

[36] Ronen Feldman, Yonatan Aumann, Amihood Amir, and Heikki Mannila, “Efficient algorithms for discovering frequent sets in incremental databases,” in *Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD’97)*, Tucson, Arizona, May 11 1997.

[37] Ramakrishnan Srikant and Rakesh Agrawal, “Mining generalized association rules,” in *Proceedings of the 21st International Conference on Very Large Databases*, Zurich, Switzerland, Sept. 23-25 1995.