Using cluster analysis to automatically thread discussion board messages

Kushal Dave (Advised by Charles Yang and Brian Scassellati)

May 6, 2002

Abstract

Although more popular than ever, web-based discussion boards are often unnavigable due to their sheer size and the inefficacy or absence of manual threading. This paper surveys existing work in document clustering and tries to identify the methods of feature selection and soft, hierarchical clustering most appropriate to this problem. Collocation, synonymy, inverse document frequency, stemming, and importance weighting are employed. An approach for maximizing relevant novelty of selected documents at each clustering level is described and compared to hierarchical baselines. A discussion from Slashdot is used as a corpus, with target categories of various granularities.

1 Introduction

One of the most popular uses of the Internet is in facilitating discussion, as demonstrated by Usenet, listservs, and, most recently, Web-based discussion boards. But as the size of the Internet community has grown, these discussions, despite splintering amongst competing sites, have reached unwieldy size. Search engines have made a growing Web more navigable for users with limited time, but no analogous tool exists for confronting the sea of conversation. Unlike search engines, which try to answer specific queries, a discussion-navigation tool should help a user sample and drill down through the available threads of conversation on a topic. While search engines focus on information retrieval, discussions call for document clustering.

A tool for clustering discussions would be useful beyond mere navigation, though. It could be used to combine two related forums into a single conversation. For example, a news site that ties discussions to individual articles might offer a merged view from related articles, so that the discourse does not have to start from scratch, losing valuable insights and making the site less useful for readers. Similarly, two different sites, each of which maintain their own proprietary discussion boards, might choose to reciprocally link conversations on related topics, so that they each retain their distinct community while infusing greater variety. Here, too, clustering could help join the threads being maintained on the separate servers.

Some popular discussion forum tools—such as WebCrossing (webx.com), used at reputable sites like nytimes.com and theatlantic.com—do not provide for threading and browsing and are generally unnavigable. If some sites are unwilling to change the linear nature of discussions, clustering could provide a critical alternative method of browsing.

Most existing work on document clustering, however, is focused on more straightforward tasks, such as sorting news articles. This is due in part to the ready availability of tagged corpora for this work, such as the Reuters corpus. Even when work is done with discussions, such as Usenet, it is done only at a macroscopic scale, using the number of messages correctly placed in their originating group as a metric.

Trying to cluster within a discussion, on the other hand, means confronting sparse data problems, spelling and grammatical errors, broad vocabularies, a need for precision and sensitivity, and abstract categorizations that are difficult to define or measure. This last element is the most important, since the application of learning algorithms is impossible or insufficient for handling this task. Furthermore, the goal of sorting discussion messages should not be to form a summary, since the opinions posted there often only make sense in context. Instead, the objective should be to help a user sample key documents and then intelligently navigate deeper into the discussion.

2 Background

2.1 Feature selection

The first step in the process of generating clusters is selecting relevant features, which are then conventionally represented as vectors of values. There are a number of ways to represent the set of words in a document and subsequently weight those representations, and these can have significant implications for the eventual performance of clustering.

2.1.1 Generalization

With very small documents, it is possible that the use of synonymous but different words can obscure significant connections. To solve this problem, databases of semantic relationships can be used to project words into a reduced space where synonymous words share a common identifier. Furthermore, if taxonomical relationships are available, words can also be mapped to higher levels of generality, where they are more likely to overlap with other terms.

WordNet from Princeton University is "an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets." It is an ideal tool for helping connect superficially unrelated documents. Among the available relations are *hypernyms* ("the generic term used to designate a whole class of specific instances") and *synsets* ("a set of words that are interchangeable in some context").

A particularly elaborate and rigorous approach to *lexical chaining* can be found in [10], where words with close relationships in WordNet are put into "chains," which are then used to produce links between news articles. WordNet has also been used for improving information retrieval in a heterogeneous corpus by finding a set of terms related to a query [9]. In general clustering, metrics such as "hypernym density," the relative frequency of a given hypernym in a document, have been used to find similarities. However, performance depends on the corpus: newspaper articles did not benefit from the technique, while song lyrics suffered from overgeneralization. Categorizing USENET messages did show improvement [20].

Stemming is another way of helping connect related words. The Porter stemmer is the most popular of these, using a set of suffix-stripping rules to recover roots. In this way, tense or conjugation of verbs does not hinder retrieval of basic meaning [18]. However, the stemmer is not perfect, and it generalizes to such a great extent that unrelated words may sometimes be given the same root.

2.1.2 Disambiguation

One side effect of using semantic tools like WordNet is a proliferation of word senses. In clustering, it is sometimes possible to just let these senses work themselves out—within a corpus, the more valid synsets will become more heavily weighted, and even those that are incorrect will really just be stand-ins for the original word. However, this is very imprecise.

Instead, it is possible to look at collocations, or pairs of words that occur together. By using a parser, pairs that are linguistically related (such as verbs and their objects), and their parts of speech, can often be determined. One study had only limited success with focusing on collocation for the purpose of *word sense disambiguation* (WSD) [15].

Collocations might still be useful in clustering even if they do not effectively perform WSD, though. Pairs of words can help qualify words enough to better identify similar passages. In fact, purely proximity-based (non-linguistic) co-occurrence features have proved helpful in clustering documents when used in conjunction with simple unigrams [7].

An equally useful tool for making words mean more is *anaphora resolution*. Identifying the referents of pronouns helps add weight to the real subject of a given portion of a document. The key to this process is centering, which provides a method for assigning the most likely (coherent) referents while iterating through a document [11]. However, centering methods offer only probable resolutions and do not seem to be well-suited to the implicit subjects, incoherence, sparse data, and knowledge-dependent pronouns frequently used in discussion boards.

2.1.3 Metadata

The use of metadata in information retrieval has been popularized by Google [4], which uses the way documents are linked to as an indicator of their relevance to a particular topic. Google also uses heuristics like giving additional weight to words occurring earlier in a document or having special formatting. In the case of a discussion board, it is possible to use the number of replies, moderator ratings, information about an author, hyperlinks contained in a message, proper nouns (named entities) in a message, the content of a parent message, quoted parts of previous messages, formatting of words in a message, chronology, and length as cues for clustering.

In a perfect world a preponderance of user- or moderator-supplied metadata would obviate the need for automated clustering processes. But this seems unlikely to occur. To be sure, there are a variety of ways in which bulletin boards attempt to better delineate the types of content contained therein. For example, some bulletin boards, such as the popular Ultimate Bulletin Board (infopop.com) allow users to rate posts or indicate the top of content being discussed. Some innovative work is also being done in the world of groupware, as discussed in a survey at (udell.roninhouse.com/GroupwareReport.html). The specific metadata available in Slashdot, our chosen corpus, is discussed below.

2.2 Similarity metrics

Once features from a set of documents have been placed in vectors, most algorithms require a way of reducing the difference between them to a number. This in turn determines the objective function of the algorithm.

2.2.1 Traditional metrics

If we envision the vectors as actually describing a p-dimensional space, where p is the number of features being considered, then a variety of metrics for measuring "distance" can be employed. Traditional norms, such as the Manhattan (L_1) and Euclidean (L_2) distance, are a relatively straightforward methods of determining the distance between two points. Conversely, the similarity of two points can be found by looking at their cosine distance (dot product of two vectors, normalized). However, these metrics make certain assumptions about the relative importance of different features, and they do not do a good job of reflecting that fact that two documents do not overlap at all.

Distances can be converted into similarities or dissimilarities by using statistical methods. Correlation coefficients can be found for documents and/or features. For more about metrics, see [13] and [5].

2.2.2 Probability metrics

We can also look at the similarity of two documents based on differences in the probabilities of each feature, weighted by the probability of the feature. The novelty of one document compared to another can be determined using information theoretic concepts such as *cross entropy* or *mutual information*.

Probability has also been used to generate *language models* from the bigrams and unigrams used in a given document. The likelihood of a given model producing a query can then be used to facilitate information retrieval [21]. However, in sparse clustering problems, it seems much more important that a term occurs in two documents than the difference in the relative frequency of each.

2.2.3 Incorporating corpus frequencies

In addition to the absolute or probabilistic weight of a term in a given document, it may also be useful to look at a term's global behavior. Identification of particularly common words, for example, can help eliminate stop words that do not contribute to the meaning of a text. Even more powerfully, weighting words inversely in relation to their frequency in the corpus can help emphasize the words most likely to indicate relevant relationships. The number of documents a given term appears in is its *document frequency*, and hence this is referred to as *inverse document frequency*. On the other hand, it may also be useful to ignore words with too low of a frequency, since they may be meaningless outliers.

Corpus frequencies can also be used as a basis for smoothing techniques that overcome sparse data problems, which in turn allows for the use of language models and other conditional probabilities. Additionally, looking at the extent to which the probability of a word in a given document deviates from the word's frequency in the corpus overall can in a sense normalize measurements of probability differences between documents.

2.2.4 Other approaches

Several metrics for clustering are more domain-specific and are not easily classified in the categories above. For example, suffix-tree clustering uses an augmented trie to keep track of documents that contain a common phrase. This has the useful property of being able to identify what documents that have been clustered together have in common. However, it can be space intensive and is only successful when the desired features are actually repeated phrases [26].

Another way at looking at clustering is to look for patterns, rather than at absolute distances. In some ways, this is similar to the approach of information theory. One study found this to be effective in biological applications [22].

Some of the best distance metrics are not metrics at all—the original data plays an active role in the data structure rather than being reduced to a number.

2.3 Clustering

Cluster analysis is an area of statistics and unsupervized learning that has a wide variety of computational applications, including for logistics, information retrieval, bioinformatics, multicasting, and text processing. A good summary of clustering is available in [16] and a more detailed treatment can be found in [13].

While all clustering involves a similar optimization problem of minimizing dissimilarity or distance (or equivalently maximizing similarity or proximity) the specific approaches vary. Several attributes are used to characterize clustering methods. *Hard clustering* assigns each object to only one group, while *soft clustering* allows probabilistic assignments to multiple clusters. *Hierarchical clustering* nests clusters within clusters, while *partitioning* is simply a flat assignment. *Supervised* algorithms can be used when a target clustering is known, otherwise only *unsupervised* methods are available.

Parameters such as threshholds or a predetermined number of clusters must often be set beforehand, although it may be possible to determine their values through guess-and-check. Applying partitioning methods recursively and/or at different granularities can be used to produce hierarchical or soft clusterings.

2.3.1 Dimensionality reduction

Not only can dimensionality-reducing operations be helpful in making clustering operations faster, they can also be viewed in themselves as a method of clustering. By finding strongly correlated documents or features, it may be possible to identify the distinct topics within a corpus.

Latent Semantic Indexing is a common dimensionality-reduction tool used in information retrieval, based on Singular Value Decomposition, and can decrease features into a smaller space. In information retrieval, SVD facilitates making search engines with practically computable term-document matrices. Principal Component Analysis, meanwhile, identifies the main directions along which documents lie. By identifying documents that are particularly strong or weak along a particular component, it is sometimes possible to identify topics in documents [17], [6], though results in areas such as biology have been less promising [25].

However, both techniques make assumptions about the relative importance and independence of terms and documents, as well as being computationally expensive. Some of these issues are dealt with in [1], which adds weight to infrequent terms or outlier documents that might otherwise be considered noise, improving the overall value of the reduced dimensions. It is also possible to fit objects to a nonlinear curve, instead of using linear methods.

2.3.2 Agglomerative clustering

Agglomerative clustering is an intuitive, greedy algorithm for producing hierarchical clusterings. After initially placing each document in its own cluster, the best-connected clusters are joined until all clusters are placed under a common parent. The metric used for joining clusters can vary: average distance, minimum distance, maximum distance are the most common.

An interesting alternative to placing each document by itself in the initial clustering is to make each initial cluster correspond to a feature (word). Each cluster contains the documents that possess that word, and clusters are joined based on the number of documents they have in common. This clever technique provided effective soft clustering of documents in a test corpus [14].

2.3.3 K-means, k-gaussians, k-nearest neighbor

Most methods of partitioning involve the selection of prospective centers for a predetermined k number of clusters, and then adjusting these centers in order to minimize some sort of coherence or entropy or expectation metric, such as the sum of distances from the center or the distance between the most distant elements in a cluster. This process is repeated until some sort of convergence or threshhold is reached. Such methods can also be applied agglomeratively, choosing to join clusters in ways that minimize dissimilarity. In the Gaussian model-based approach, a multivariate probability distribution is created for each center, specifying the extent to which a document might be associated with a particular center. The biggest failing of such methods is that the chosen k may be incorrect in terms of the real data, even if numerically it appears to be working well. Also, each specific variant has a bias towards certain types of clusters. But the methods are straightforward and popular and have been used successfully for a variety of applications.

Fuzzy variations of these algorithms assign each object a probability of belonging to nearby groups based on their relative proximity. There are a variety of ways of modeling these relationships, but the ability to create soft partitions that naturally assign

likelihoods to membership in each group is relatively unique.

2.3.4 Graph-theoretic approaches

The clustering problem can also be easily envisioned as an attempt to find groups in a graph, where objects are vertices and edge weights or the presence of an edge indicate similarity. Then, the goal can be to recursively divide the graph in ways that yield the minimum flow between the two partitions (max-cut) [24]. Or it can be viewed as an attempt to locate cliques or near-cliques [3]. It is also possible to envision partitioning as a coloring problem, where edges connect dissimilar object vertices, which in turn is easily construed as a vertex cover problem. A cheaper graphical approach is to impose a tree on the elements using a greedy maximal or minimal spanning tree algorithm. These methods do not always produce coherent clusters in the way one might desire, though, and they can only be readily used for hard clustering.

2.3.5 Optimizations

Because the dissimilarity-minimization optimization problem is often NP-complete, a variety of methods are used in order to make it more manageable. *Buckshot* involves randomly choosing a set of points with which to form the core of clusters, and the remaining points are just added to the most appropriate clusters. *Fractionation* relies on finding clusters within randomly chosen subsets of the graph, and then attempts to combine these clusters. [5]

A second method of optimization is to improve an initial random assignment using exchanges, until the assignment converges. This must be repeated multiple times to avoid settling into local maxima. Columbia's SIMFINDER is one of many approaches that uses this method [12].

2.4 Presentation

Part of designing or choosing clustering and feature selection algorithms is knowing how the information will be eventually used. It can influence whether microscopic or macroscopic similarities are considered, whether hard or soft methods are used, and whether hierarchical methods are used.

2.4.1 Information retrieval

One of the definitive surveys of hierarchical clusters [23], uses only hard methods and is aimed at improving search results by identifying relevant clusters of documents. The hierarchy can then be traversed downward or upward until a certain criteria is met. Because doing so requires the selection of representative documents at each level of the *dendrogram*, it provides some insights for clustering that only intends to provide a method of browsing.

2.4.2 Summarization

Much of the current work in clustering in language processing is focused on summarization, particularly of news articles. Notably, the NewsBlaster project at Columbia has produced some impressive results [12]. Summarization is a very different problem from trying to cluster messages for browsing; for summarizing, smaller units are being clustered, and the definition of similarity is much more straightforward and specific.

2.4.3 Discourse theory

An optimal clustering tool for discussions would go beyond merely identifying related documents. Instead it would be able to identify the relationships between documents. In *cross-document structure theory*, a variety of potential relations, such as citation, summary, and modality are delineated [19]. CST is an extension of discourse theory, which provides methods for parsing documents in order to derive intra-document relationships, such as elaborations, evidence, and contradictions.

Such formal linguistic theories also have relatives in sociological and psychological observations of discussions. Effective clustering techniques for discussion boards might even be helpful in furthering such studies. However, outside of rigid corpora such as news articles, it seems unlikely automated methods will be able to recover these relationships without additional knowledge.

2.4.4 Visualization

The problem of actually projecting a high-dimensional space into two dimensions is a very difficult one, and complex techniques like self-organizing maps are needed. Although this may sometimes allow identification of clusters through visual inspection, one benefit of cluster analysis is that it helps make visualization more feasible by breaking the set into distinct categories.

One way around the visualization problem is to merely display documents relative to the one currently being viewed. This is the approach of [2],

which shows documents along a line, spaced relative to their distance from the one currently being viewed.

2.4.5 Linking

Instead of concentrating on depictions, it may be more useful to think about how a user will want to navigate from document to document. Navigation of hierarchies via successive screens or a collapsible tree is straightforward to imagine. One successful approach is the Tilebars method, which is part of Scatter-Gather [5]. The Scatter-Gather approach, which dynamically produces partitionings as a user selects increasingly finer levels of granularity, seems particularly well-suited to problems of soft, hiearchical document categorization.

Alternatively, it may be useful to try to link key paragraphs or phrases in a given document to related documents [10].

3 Approach

3.1 Corpus and Objective

Slashdot (slashdot.org) is a discussion board where computer nerds post links to and brief summaries of the latest news. These postings serve as starting points for discussions among the hundreds of thousands of users than range from around a hundred to more than 700 comments each. Randomly chosen moderators use points to moderate message ratings up and down and label them with categories such as Funny or Insightful. The system supports complete threading, but users often choose to post at the top level of discussion even when it is inappropriate, perhaps because they feel more people will see their message there, or because they did not bother to read all of the preceding discussion. Whatever the reason, these and other misplaced messages cry out for rethreading, which can be accomplished by hierarchical clustering. This task can be be separated out into a micro-scale objective—putting documents that discuss the same limited topic or offer virtually identical information close together—and a macro-scale one—putting documents discussing the same broad subject or issue together. In many ways, the goal is like that of summarization, to increase "relevant novelty" [8].

For the purposes of this project, I selected a discussion about Microsoft's Hailstorm that ended up including threads on distributed storage mechanisms, alternatives to Hailstorm, and rants about privacy. There were 152 messages that had not been designated as entirely irrelevant at the time of download. Although hardly exact and certainly not appropriate for a supervised learning approach, topics of various granularities were assigned to the documents in order to offer something better than qualitative evaluations. Table 1 lists the tags and their frequencies and Figure 1 shows how they overlap. The diagram is particularly illustrative of why we choose the approach that we do.

The decision about whether a message belongs to a given category is less than perfect, and it should simply be used as a guideline. In fact, topics that overlap a great deal are an indication that other messages not explicitly denoted as overlapping might very well do so in someone else's assessment, and that the eventual categorization should not only do a good job placing documents in the same category near each other, but should, and perhaps cannot avoid, placing documents in related categories near each other as well. This is why our eventual solution uses soft clustering.

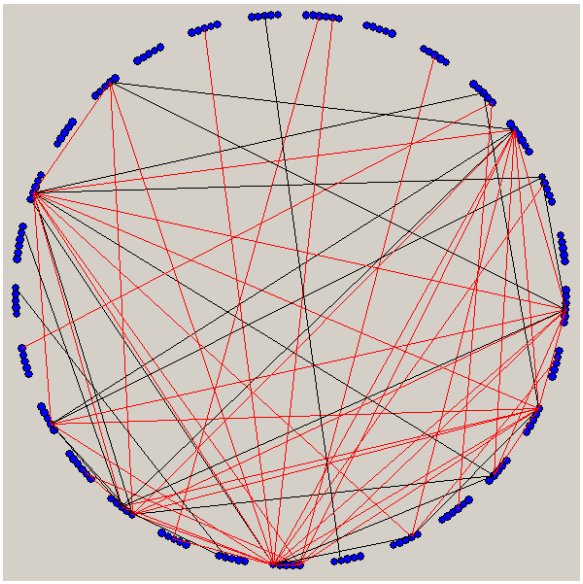

Figure 2: The strongest links in the corpus, from a prototype visualization tool.

3.2 Features

In order to make decisions about constructing feature vectors and distance metrics, both qualitative and quantitative results were considered. Although time did not permit trying all possible permutations, it seemed that the simple dot product of collocations with single-height Wordnetting and inverse document frequency weighting provided the best results.

Any further use of Wordnet led to overgeneralization and unwieldy, large datasets. Even when using

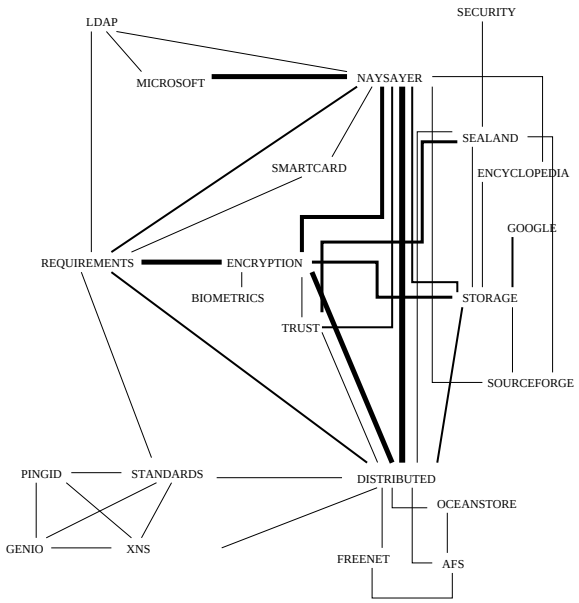

Figure 1: The overlap between manual tags.

Frequency	Tag	Common words
57	naysayer	I, my, information, repository
25	encryption	data, you, that, encrypt, key
20	distributed	client, distributed, data, information
17	requirements	data, should, information
16	storage	oceanstore, afs, store, file, distributed
12	onumber	number, control
9	trust	generation, trust
9	sealand	sealand, public
8	microsoft	hailstorm, microsoft, company
6	freenet	freenet, node, data, around
5	standards	institution, xns, protocol, standard
5	smartcard	card, smart
4	afs	afs, oceanstore, store
3	security	sealand
3	google	google, archive, usenet
3	biometrics	authentication, biometrics, token
3	oceanstore	afs, oceanstore, store, around, file
2	sourceforge	public
2	genio	pingid, xns, genio
2	ldap	ldap
2	book	related
2	encyclopedia	encyclopaedia, earth
2	XNS	xns
2	pingid	pingid, xns, genio

Table 1: This is the set of manual tags chosen for the corpus, along with their frequency and representative words. Many documents have multiple tags. Words are selected from the unigrams with the strongest IDF-weighted total strength in the documents of the given category.

only one level of hypernyms, it was possible to group together words such as "data" and "information" or "take" and 'accept."

Each feature was given equal weight, although those occurring in the title are given double weight, and the initial sentences are given progressively less. Words were put together in linguistically-meaningful collocations using MINIPAR, which also enabled discovery of parts of speech for querying Wordnet.

The benefits of Wordnet are purchased at a price. 1,316 unigrams multiply into 5,675. Similarly, when collocation is added, these become 44,559 features—even after the threshhold is applied.

Examples of relations discovered by the parser include adjective-noun ("centralized repository"), noun-noun ("information repository"), subject-verb ("user demand"), determiner-noun ("the need"), and object-verb ("demand convenience"). A side effect of this method is that words that appear in more than one collocation are more heavily weighted than those that appear in one or none. A more equitable system of global weighting could not be settled on.

The original collocation (after stemming and case elimination) received 50 percent of the weight, and the many possible permutations from the potential Wordnet meanings of both words divided the remaining weight equally. It was left up to the repeated occurrence of similar collocations to effectively provide disambiguation of the word. Figure 3 provides an overview of the process.

3.3 Vectors

Inverse document frequency was computed as $w_i = \log \frac{N}{df_i}$ where N is the number of documents in the corpus and df_i is the document frequency. Not using inverse document frequency would have led to unreasonable bias towards larger documents. Numerous overlapping, common words could make two documents look related, even if they have no more in common than any arbitrary pair from the corpus.

This is not helped by lack of a preset list of *stopwords* like "it" or "a". However, because of the use of collocations and IDF, these words are ignored in their unigram form and can help qualify the usage of content words—for example, "my database" versus "the database."

In order to constrain the size of the data, and to ensure that genuinely content-less words were ignored, a threshhold was established—any collocations or unigrams occurring in more than two-thirds of the documents were discarded, as were features occurring only once. An additional optimization tracked WordNet features that appeared as the result of only one word. These were then coalesced in a reduced vectors file—a sort of zero-information-loss dimensionality-reduction.

3.4 Distances

Methods that were overly sensitive to the relative frequency of a term in a document, such as entropy or normalized cosine distances, were not very successful, because very short, irrelevant outlier messages produced strong links to other documents on the basis of single words. The mere presence of common words in two documents in such a sparse data set is more important than their relative importance in each document, and this is the insight behind just using what is known as the *simple matching coefficient c*, the number of terms two documents have in common.

In order to include some notion of weight, the dot product of feature vectors is used to calculate similarities ($s(v_i, v_j) = \sum_k v_{i,k} * v_{j,k}$). Although larger documents do receive larger absolute weights, methods were developed during clustering to use as well as compensate for this. For a comparison of feature selection and distancing methods, see Table 2.

3.5 Clustering

Our goal in this clustering is to identify a soft, hierarchical method of sorting these messages. The method should also provide a direct way of representing the contents of a given cluster, such as a key message or relevant key words.

3.5.1 Initial attempts

As a baseline, a rudimentary average link agglomerative clustering algorithm is implemented. Although far from optimal, the baseline does a better-than-random job of clustering the groups.

Our observations of the corpus then suggest an improvement: due to the lack of normalization, the highest-valued links tend to be between the highest-valued documents, which also happen to be the longest ones. And we found that standard normalization is not a good solution to this problem. But some way of indicating how important a given link is to a particular document would still be helpful. We therefore calculate a second measure—the average of the percent of the total link strength of the objects at each each end of a given link that this link represents. This is equivalent to the *Dice coefficient*, which is defined as $\frac{2c_{i,j}}{s_i+s_j}$ where $c_{i,j}$ is the similarity between

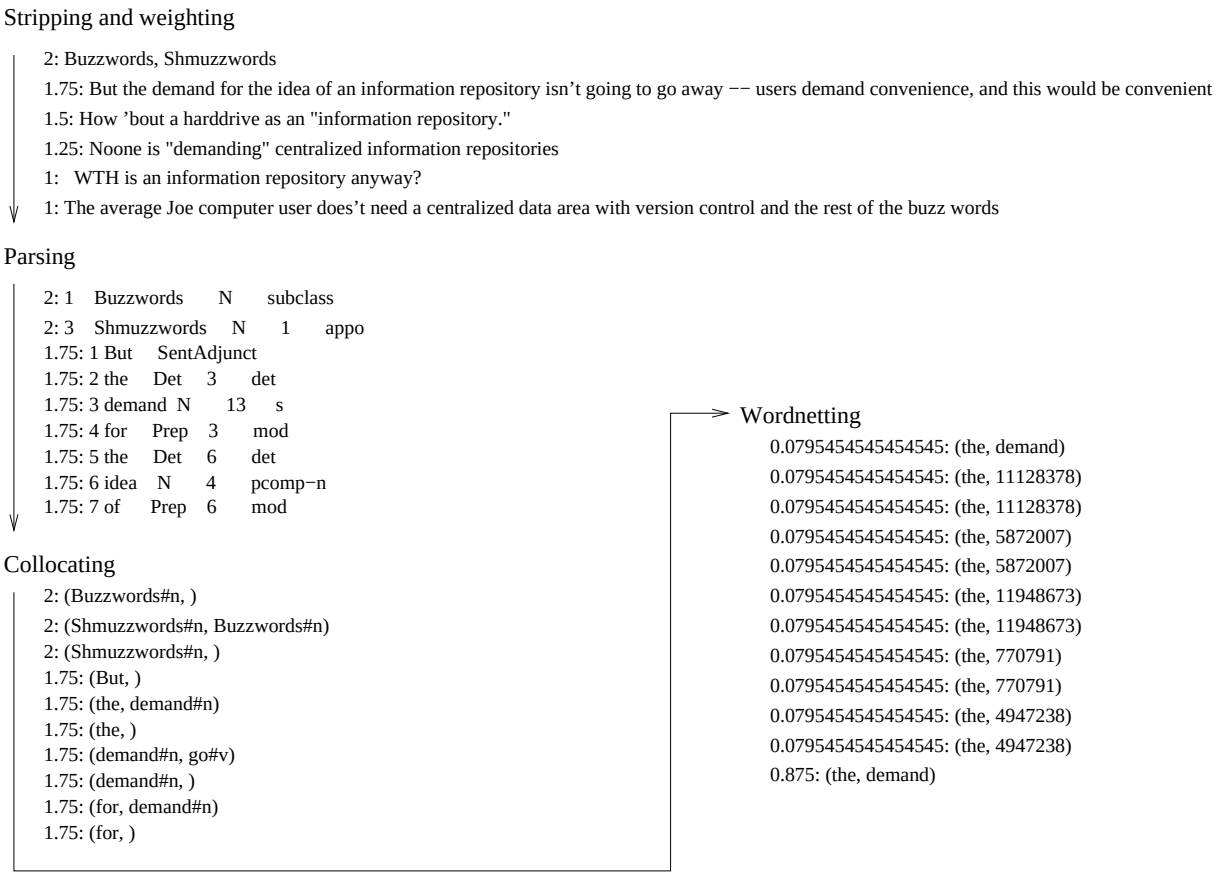

Figure 3: How features are selected. The resulting collocations and weights are then used to construct feature vectors and compiled into a matrix.

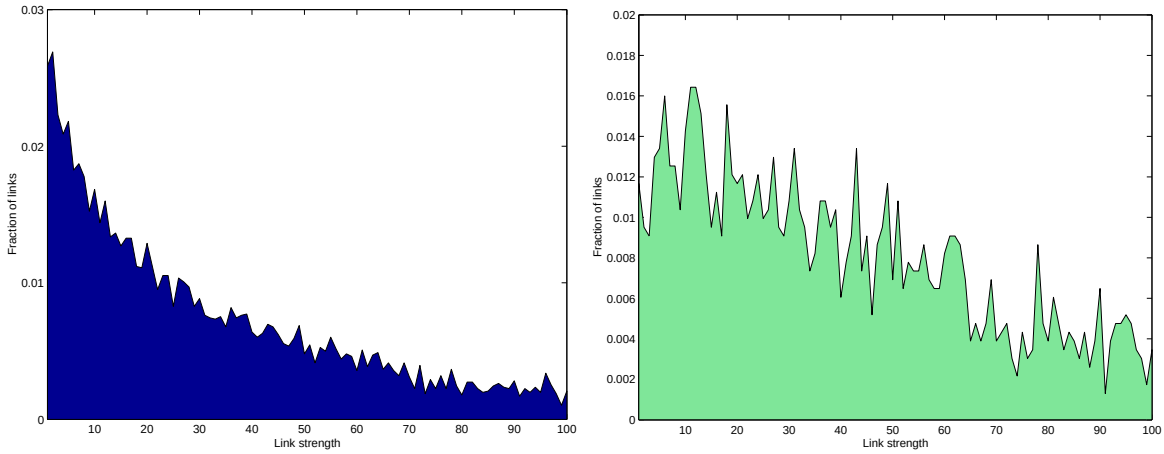

Figure 4: Although averages do not tell the full story of the distribution, it is clear that the default feature selection method produces in-group links (right) that are generally stronger than those in the overall set (left). (Links less than 2 or greater than 100 not shown.)

IDF	Dampening	Normalization	Collocation	Wordnet	Link quality
Y	N	N	Y	h=1	126.41
Y	N	N	N	h=1	122.53
Y	N	N	N	h=2	122.39
Y	N	N	Y	N	118.51
Y	log	N	Y	h=1	108.29
Y	sqrt	N	Y	h=1	106.33
Y	N	N	N	N	104.42
N	N	N	Y	h=1	75.82
Y	N	Y	Y	h=1	-43.99

Table 2: Effects of feature selection on links. "Link quality" refers to the percent by which the mean similarity between members of the manually-determined categories exceeds the mean similarity of the corpus as a whole.

two objects and $s_i = \sum_j c_{i,j}$. It incorporates some of the effects of entropy-based approaches to distance by looking at the relative importance of a given similarity and helps counteract the tendency for certain documents to be "popular" simply due to length.

We try agglomeratively clustering based on the highest-percentage average links, rather than those with higher absolute values. This shows some improvement in grouping both general and specific categories. We also try a maximum spanning tree based on percentage but rooted at the vertex with the highest absolute value, and this performs slightly better. Although this method actually misses some of the more specific categories, it does much better at keeping general ideas together. For a comparison of methods, see Table 3.

3.5.2 New soft algorithm

Our new approach tries to avoid any explicit distance metric and instead tries to focus on the features themselves, while building on the lessons from the baselines. At each level, our goal is to identify the minimum number of documents to represent or "cover" a maximal portion of the features that subtree is intended to represent.

To begin with, each document is assigned a certain "novelty" rating based on the IDF-weighted sum of its features. As the most novel available document is added to the set of documents at the root, the value of features it contains is decreased in the novelty of the remaining documents. The invariant with each search for a member to add is the following:

$$coveredness_j = \sum_{k \in chosen} v_{k,j}$$

$$novelty(v_i) = \sum_{j=0}^{N} \frac{v_{i,j}}{coveredness_j}$$

Attempts to cover the set of desired features terminates when a point of diminishing returns is reached, so that the increase in the sum of the coveredness from a new document is less than a certain percentage.

The goal of child levels is to best cover their parent, and this method is applied recursively until all documents have been included. We now need to redefine novelty, and we reuse the effective idea of percentages indicating importance.

$$novelty_{p_i}(v_j) = \sum_{k=0}^{N} \frac{\frac{v_{i,k}+v_{j,k}}{\sum_l v_{i,l}+\sum_l v_{j,l}}}{coveredness_k}$$

The process is otherwise identical to the technique employed above.

This method does not appear to do as well as the naive hard clustering methods based on our simple numerical measure (Table 4). However, subjectively, it is more useful because it provides easy ways of representing the content of a given subtree, and it does fairly well at putting related documents close together.

4 Evaluation

4.1 Constraints and parameters

The soft clustering technique may still benefit from fine-tuning various arbitrary criteria and weights.

For example, in order to encourage the selection of new documents and avoid circularity, documents can only be repeated within the same generation—after that, they may not be used again. This leads to a sort of weak softness, but still works fairly well. In practice, it may be desirable to do this dynamically, as in Scatter-Gather, and to eliminate from consideration

documents a user has already seen. Still, trying different constraints may produce better static results.

Different performance could also result from different criteria for terminating attempts to cover a given target. This could mean different percentages, or a wholly new way of deciding to stop, even a fixed number of children at each level.

Finally, different methods of weighting novelties and the features they derive from could be tried. Not using percentages to normalize novelty was already tried and discarded, but dampening may prove helpful.

4.2 Computational complexity

When a document is added to the corpus in the soft clustering algorithm, the novelty of the remaining documents must be updated. A novelty calculation is in turn linear in the number of features. Therefore, the algorithm is a disappointing $O(N^2M)$ where N is the number of documents and M is the number of features, which usually far exceeds N. Some optimizations may be possible, such as only updating novelties after selected threshholds are passed.

The algorithm is more reasonable in space, requiring an N x M array for the features, plus some additional arrays linear in one of these two dimensions.

Both of these requirements are in line with the requirements of baseline techniques and other clustering methods, particularly those that are fully deterministic.

4.3 Performance Metric

A variety of metrics are available to us, though none of them are wholly satisfying. They generally lie on a gradient of whether feature selection or clustering is more influential of the outcome, though this is also influenced by which independent variables are manipulated.

We employ a method that lies somewhere in the middle. By manually tagging the corpus which categories of varying specificity—the broadest category covers around a third of the messages, the smallest ones covering only two messages—we can see how the average distances in these groups are affected by changes to both feature selection and clustering.

The average distance between any two given points in the graph is compared to the average distance between any two given points in a certain group. The percentage by which the second number exceeds the first is a link quality score.

For examining feature selection and distance metrics, the actual distances between the documents serve as the distances for this comparison. For trees, the sum of the number of edges (generations) from each of the two documents to their deepest common parent is considered to be the distance. For softly-clustered trees, only the minimum value is used, although the average was also tried.

Method	Link Quality
Naive hierarchical	6.52
Percentage hierarchical	16.55
Percentage max. spanning tree	16.79
Soft covering method	7.80
Soft covering method with percentage	10.44

Table 3: Relative efficacy of clustering methods. Link quality here refers to the percent by which the average link within a cluster is shorter than the average link in the corpus as a whole. Distances are the sum of the distances from each document to their deepest common ancestor. For soft clustering, the closest distance is used.

Of course, the categories are inexact, and the percentages are not perfect comparisons since the distributions being considered are not normal. However, they do supplement subjective observations that performance is improving.

5 Conclusion and future work

The problem of clustering documents is a difficult one, and the problem of clustering discussions is worthwhile, but even more difficult. Appropriate feature selection has been shown to provide better-than-random associations between related messages. Finding a clustering algorithm capable of generating a good soft hierarchy is a problem that has not been dealt with extensively, and certainly not in this context. The "covering" method makes intuitive sense and has provided some promising, if not definitively better, results.

There are also many broader avenues for future work on the problem:

1. *Computational structure* In real-world applications such as those suggested in the introduction, it may be important to find ways for document clustering to run on-line (incrementally or in reasonable time) and in a distributed fashion (allowing multiple servers to aggregate their data).

2. *Goal clarification* The optimal interface for navigating discussions remains undiscovered, and it

Category	Freq.	Percentage MST				Soft covering			
		Average		Median		Average		Median	
		Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.
OVERALL		7.6		7		1.2		1	
naysayer	56	6.80	-10.03	6.00	-14.29	1.30	5.46	1.00	0.00
encryption	24	5.73	-24.18	4.00	-42.86	0.97	-21.16	1.00	0.00
distributed	19	8.12	7.46	7.00	0.00	1.23	-0.67	1.00	0.00
requirements	16	3.47	-54.08	3.00	-57.14	1.22	-1.13	1.00	0.00
storage	15	7.68	1.67	6.00	-14.29	0.86	-30.47	0.00	-100.00
onumber	11	2.17	-71.33	1.00	-85.71	1.08	-12.86	0.00	-100.00
trust	8	12.03	59.15	12.00	71.43	1.42	14.75	1.00	0.00
sealand	8	12.67	67.61	17.00	142.86	1.08	-12.25	0.00	-100.00
microsoft	7	4.86	-35.73	5.00	-28.57	1.00	-19.00	1.00	0.00
freenet	5	2.73	-63.83	1.00	-85.71	0.13	-89.20	0.00	-100.00
standards	4	5.80	-23.25	7.00	0.00	1.60	29.60	1.00	0.00
smartcard	4	1.00	-86.77	1.00	-85.71	1.10	-10.90	1.00	0.00
afs	3	0.67	-91.18	0.00	-100.00	0.33	-73.00	0.00	-100.00
security	2	16.00	111.71	13.00	85.71	0.00	-100.00	0.00	-100.00
google	2	0.33	-95.59	0.00	-100.00	0.00	-100.00	0.00	-100.00
biometrics	2	0.00	-100.00	0.00	-100.00	2.33	89.00	3.00	200.00
oceanstore	2	5.00	-33.84	7.00	0.00	0.33	-73.00	0.00	-100.00
sourceforge	1	22.00	191.11	22.00	214.29	0.00	-100.00	0.00	-100.00
genio	1	0.00	-100.00	0.00	-100.00	2.00	62.00	2.00	100.00
ldap	1	3.00	-60.30	3.00	-57.14	3.00	143.01	3.00	200.00
book	1	17.00	124.94	17.00	142.86	0.00	-100.00	0.00	-100.00
encyclopedia	1	0.00	-100.00	0.00	-100.00	0.00	-100.00	0.00	-100.00
XNS	1	0.00	-100.00	0.00	-100.00	0.00	-100.00	0.00	-100.00
pingid	1	0.00	-100.00	0.00	-100.00	2.00	62.00	2.00	100.00
AVERAGE	195		-16.79		-19.41		-10.44		-22.56

Table 4: Detailed comparison of clustering methods. Although the soft clustering algorithm outperforms the best hard baseline in some categories, its overall performance is still not as good by this measure. Of course, the soft clustering metric only only uses the minimum distance between two points, and thus does not provide an exact comparison.

is therefore unsurprising that the goals of message clustering remain ambiguous. For example, it may be useful to look at paragraphs or sentences rather than full documents and just create local inter-document links. There is also some question as to whether it is desirable or feasible to cluster by both opinion and topic.

3. *Linguistic tools* Improved linguistic tools would be very useful. For example, it would be nice to have a way of identifying the connotations of language being used: does the piece use invective? laudatory language? neutral, factual words? Additional linguistic tools to identify key named entities and correct spelling errors would be helpful.

4. *Community understanding* Identifying the types of writers and communities in writers, as well as the tendencies of moderators, could help make better use of available metadata in the discussion. Those authors known to be like-minded could be grouped together, and those known to be inflammatory or off-topic could be buried deep in the hierarchy.

5. *Discussion heuristics* There seem to be a variety of potential heuristics for clustering discussions that could be explored in further detail. As mentioned earlier, looking at the number of replies, threading, links to other web sites, and even length can be quite helpful.

References

[1] R. Ando. Latent semantic space: Iterative scaling improves inter-document similarity measurement, 2000.

[2] R. Ando, B. Boguraev, R. Byrd, and M. Neff. Multi-document summarization by visualizing topical content, 2000.

[3] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[4] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[5] Douglass R. Cutting, Jan O. Pedersen, David Karger, and John W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329, 1992.

[6] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

[7] Eleazar Eskin, Judith Klavans, and Vasileios Hatzivassiloglou. Detecting similarity by applying learning over indicators.

[8] Jade Goldstein, Vibhu O. Mittal, Jaime G. Carbonell, and James P. Callan. Creating and evaluating multi-document sentence extract summaries. In *CIKM*, pages 165–172, 2000.

[9] Julio Gonzalo, Felisa Verdejo, Irina Chugur, and Juan Cigarran. Indexing with wordnet synsets can improve text retrieval. In *Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP*, pages 38–44, Montreal, Canada, 1998.

[10] Stephen J. Green. Building hypertext links in newspaper articles using semantic similarity. In *Proceedings of Third Workshop on Application of Natural Language to Information Systems (NLDB '97)*, pages 178–190, Vancouver, June 1997.

[11] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225, 1995.

[12] Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen R. McKeown. Simfinder: A flexible clustering tool for summarization.

[13] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, New York, NY, 1990.

[14] Ravikumar Kondadadi King-Ip Lin. A word-based soft clustering algorithm for documents. In *Proceedings of 16th International Conference on Computers and Their Applications*, 2001.

[15] D. Lin. Using collocation statistics in information extraction, 1998.

[16] Christopher D. Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[17] Charles K. Nicholas and Randall Dahlberg. Spotting topics with the singular value decomposition. In *PODDP*, pages 82–91, 1998.

[18] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[19] D. Radev. A common theory of information fusion from multiple text sources, 2000.

[20] Sam Scott and Stan Matwin. Text classification using WordNet hypernyms. In Sanda Harabagiu, editor, *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, pages 38–44. Association for Computational Linguistics, Somerset, New Jersey, 1998.

[21] Fei Song and W. Bruce Croft. A general language model for information retrieval (poster abstract). In *Research and Development in Information Retrieval*, pages 279–280, 1999.

[22] Haixun Wang, Wei Wang, Jiong Yang, and Philip S. Yu. Clustering by pattern similarity in large data sets.

[23] Peter Willett. Recent trends in hierarchic document clustering: A critical review. *Information Processing & Management.*

[24] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.

[25] K. Yeung and W. Ruzzo. An empirical study on principal component analysis for clustering gene expression data, 2001.

[26] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In *Research and Development in Information Retrieval*, pages 46–54, 1998.