A Probabilistic Approach to Full-Text Document Clustering

Moises Goldszmidt
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
moises@erg.sri.com

Mehran Sahami
Gates Building 1A
Computer Science Department
Stanford University
Stanford, CA 94305-9010
sahami@cs.stanford.edu

Abstract

To address the issue of text document clustering, a suitable function is needed for measuring the distance between documents. In this paper we explore a function for scoring document similarity based on probabilistic considerations: similarity is scored according to the expectation of the same words appearing in two documents. This score enables the investigation of different smoothing methods for estimating the probability of a word appearing in a document, for purposes of clustering. Our experimental results show that these different smoothing methods may be more or less effective, depending on the degree of separability between the clusters. Furthermore, we show that the cosine coefficient widely used in information retrieval can be associated with a particular form of probabilistic smoothing in our model. We also introduce a specific scoring function that outperforms the cosine coefficient and its extensions, such as TFIDF weighting, in our experiments with document clustering tasks. This new scoring is based on normalizing (in the probabilistic sense) the cosine similarity score, and adding a scaling factor based on the characteristics of the corpus being clustered. Finally, our experiments indicate that our model, which assumes an asymmetry between positive (word appearance) and negative (word absence) information in the document clustering task,

outperforms standard mixture models that weight such information equally.

1 Introduction

As the amount of on-line information continues to grow at an ever-increasing rate, the need for tools to help manage this information also rises. One such tool is the capability to cluster documents of similar content, to aid in both the retrieval of information and its presentation to the user. Early work in information retrieval (IR) stressed the use of clustering as a means of improving the ability to find documents relevant to a query [vRJ71] [Sal71]. This work was based on the *Cluster Hypothesis* [vR79], which states that "closely associated documents tend to be relevant to the same requests." With this as a working assumption, document collections could be clustered a priori, and then new queries could simply be matched against clusters rather than against each document individually. Such cluster-based matching could speed the retrieval process and possibly find relevant documents that do not explicitly contain the words in the user's query.

More recently, applications of document clustering such as Scatter/Gather [CKPT92] [HP96] have been used to enable entire collections and query retrieval results to be browsed more easily. Work in this area has shown that document clustering is often an effective way to give the user a better sense of the topics present in a set of documents [PSHD96].

The success of such systems often hinges on the effectiveness of the clustering methods employed. There is a long history of empirical work in document clustering, an excellent survey of which is given by Willett [Wil88]. Indeed, the description of Scatter/Gather is very specific about the clustering methods used, reflecting the years of comparative work in the IR community that continues today [SS97].

While empirical work in document clustering has advanced the state of the art in performance, no equivalent advancement in theoretical analysis explains why the methods arrived at through experimentation work as well as they do. In this paper, we seek to provide a foundational analysis of document clustering with the tools of probability theory. In this way, we can formalize the assumptions and models used in document clustering. Our objective is to gain new insights into the effectiveness of current clustering algorithms as well

as to open the door to improved, well-founded extensions. Uncovering the explicit distributional assumptions made in many text clustering algorithms has prompted us to investigate issues such as the treatment of evidence and different approaches to density estimation. Consequently, in this paper we propose a probability-based score for document overlap that outperforms traditional IR methods in our experiments on text clustering.

In general terms, the clustering problem consists of finding groups of data points that possess strong internal similarities. The problem is not formalized until we define what is meant by similarity. In practice, this formalization involves two separate issues: first, how one should measure similarity between data samples, and second, how one should evaluate a partitioning of a set of samples into clusters. Working in the context of document clustering, we propose a probabilistic score for measuring similarity between documents and evaluating clustering partitions.

In this context, and more generally throughout IR, a commonly used measure of similarity is obtained by representing documents as normalized vectors and then computing the inner product to find the cosine of the angle between the vectors. This measure of similarity is generally referred to as the *cosine coefficient* [Sal71]. Each dimension of the vector corresponds to a distinct word in the union of all words in the corpus being clustered. A document is then represented as a vector containing the normalized frequency counts of the words in it. Intuitively, this measure tries to capture the degree of word overlap between two documents.

On similar grounds, we investigate a probabilistic function for document overlap that scores the expectation of the same words appearing in two documents. This score prompts the investigation of different smoothing methods for estimating the probability of a word appearing in a document. As our empirical evaluation shows, different smoothing methods may be more or less effective, depending on the degree of separability between the clusters. We also show that the widely used cosine coefficient can be associated with a particular form of probabilistic smoothing in our framework. Moreover, this analysis reveals a scaling factor, given by the inverse of the probability of a word appearing in the corpus, that, when combined with our probabilistic similarity score, yields a clustering method that outperforms those based on the cosine coefficient and TFIDF weighting [SB87] in our experiments. Finally, we also experiment with alternative probabilistic approaches based on mixture models such as AutoClass [CKS+88], showing that they generally

produce inferior results.

We point out that the probabilistic score we present can easily be extended to include more sophisticated notions of document overlap, based on equivalence classes of words (e.g., synonyms), phrases, or, in general, any function on groups of words in the corpus. In this way, our score can cleanly capture the full generality of *probabilistic indexing* [Fuh89] techniques used in other contexts. Moreover, the parameters defining the contributions of different words or functional characteristic of the documents to the overall similarity score in these cases can be learned directly from the data. Finally, it should be clear that another advantage of a probabilistic score is the possibility of cleanly fusing information coming from different modalities (such as video and audio) into similarity scores over multimedia domains. These issues are the focus of our current research.

2 Probabilistic Document Overlap

To formalize the problem of document clustering, we first need to explicitly define a notion of similarity between documents. The similarity function that we will use for clustering will be based on establishing the degree of overlap between pairs of documents. To this end, we will assume that each document imposes a multinomial distribution over the set of words in the corpus. Each document doc_i is associated with an n-dimensional feature vector d_i. Each dimension of this vector corresponds to a distinct word in the union of all words in the corpus. The value of the jth component of the vector is the number of times the word corresponding to this component appears in the document. Thus, this vector representation of documents provides the sufficient statistics for computing the expected overlap between any given pair of documents. Let doc_i and doc_j be two documents in a corpus D. We will then compute the expected overlap between doc_i and doc_j in terms of the corresponding vectors d_i and d_j. We denote this expected overlap measure as $EO(d_i, d_j, D)$ and compute it as follows:

$$\sum_{w \in d_i \cap d_j} P(Y_i = w | d_i, M) \cdot P(Y_j = w | d_j, M) \ , \tag{1}$$

where $Y_i = w$ denotes the event that a word selected from document doc_i is equal to w. M, the model, contains information about the corpus D, including the total number of times each word appears in the corpus, as well as information about the partitioning of documents into clusters.

This equation is intuitively appealing. It says that the overlap between two documents i and j can be computed by estimating the probability that each word appears in each document, and then multiplying these results. As will be seen shortly, the way this probability is estimated will greatly influence the results of clustering. We focus on the different ways of estimating this probability from the statistics in each vector d_i and M, as well as the relationship of this equation to the cosine coefficient [Sal71] below. We first provide a derivation of Equation 1.

2.1 Deriving the Probabilistic Overlap

Here we investigate one possible derivation of Equation 1 and reveal its underlying assumptions. We start by defining the expected degree of overlap between two documents doc_i and doc_j in the corpus D, using the corresponding vectors of word statistics d_i and d_j. This definition is given by

$$\sum_{w \in W} P(Y_i \in d_i, Y_j \in d_j, Y_i = w, Y_j = w | d_i, d_j, M) \ , \tag{2}$$

which can be rewritten as

$$\sum_{w \in W} \begin{array}{c} P(Y_i \in d_i, Y_j \in d_j | Y_i = w, Y_j = w, d_i, d_j, M) \cdot \\ P(Y_i = w, Y_j = w | d_i, d_j, M) \ . \end{array} \tag{3}$$

The event $Y_i \in d_i$ denotes whether the word assigned to Y_i appears in d_i (i.e., has a nonzero count).

The events $Y_i \in d_i$ and $Y_j \in d_j$ are clearly independent when conditioned on Y_i, Y_j, and the vectors of statistics d_i and d_j. Moreover, the value of $Y_i \in d_i$ depends only on the choice of Y_i, and d_j within a given model M:

$$\sum_{w \in W} \left(\begin{array}{l} P(Y_i \in d_i | Y_i = w, d_i, M) \cdot \\ \quad P(Y_j \in d_j | Y_j = w, d_j, M) \cdot \\ \quad\quad P(Y_i = w, Y_j = w | d_i, d_j, M) \end{array} \right) \ . \tag{4}$$

Note that $P(Y_i \in d_i | Y_i = w, d_i, M)$ and $P(Y_j \in d_j | Y_j = w, d_j, M)$ are simply indicator functions that limit the set of words that contribute to the sum to only those $w \in d_i \cap d_j$. This reduces the sum above to

$$\sum_{w \in d_i \cap d_j} P(Y_i = w, Y_j = w | d_i, d_j, M) \ . \tag{5}$$

By applying Bayes Theorem, we find that the summation in Equation 5 is equal to

$$\sum_{w \in d_i \cap d_j} \frac{\left(\begin{array}{r} P(d_i, d_j | Y_i = w, Y_j = w, M) \cdot \\ P(Y_i = w, Y_j = w | M) \end{array} \right)}{P(d_i, d_j | M)} \ . \tag{6}$$

We make the assumption that, given the information in M, documents are independently distributed so that the statistics about different documents are independent of each other: $P(d_i, d_j | M) = P(d_i | M) \cdot P(d_j | M)$.

From probability theory we can write

$$\begin{array}{l} P(d_i, d_j | Y_i = w, Y_j = w, M) = \\ P(d_i | d_j, Y_i = w, Y_j = w, M) \cdot P(d_j | Y_i = w, Y_j = w, M) \ . \end{array} \tag{7}$$

Note that any probabilistic dependence between d_i and d_j as the result of $Y_i = w$ and $Y_j = w$ must be captured through the effect of each single word w. Since we believe that this effect is small, especially given that documents are made up of many distinct words, we make the approximation that

$$\begin{array}{rcll} P(d_i | d_j, Y_i = w, Y_j = w, M) & & & \\ \quad \approx & P(d_i | Y_i = w, Y_j = w, M) & & (8) \\ \quad = & P(d_i | Y_i = w, M) \ . & & (9) \end{array}$$

Substituting this approximation into Equation 7 yields

$$\begin{array}{rcll} P(d_i, d_j | Y_i = w, Y_j = w, M) & & & \\ \quad \approx & P(d_i | Y_i = w, M) \cdot P(d_j | Y_j = w, M) \ . & & (10) \end{array}$$

Our final assumption will be that, given the statistics of the corpus (which are part of the model M), the probability of drawing a given word from two different documents are independent events. Hence,

$$\begin{array}{rcll} P(Y_i = w, Y_j = w | M) & & & \\ \quad = & P(Y_i = w | M) \cdot P(Y_j = w | M) \ . & & (11) \end{array}$$

Substituting Equations 10 and 11 into Equation 6 yields

$$\sum_{w \in d_i \cap d_j} \frac{\left(\begin{array}{r} P(d_i|Y_i = w, M) \cdot P(d_j|Y_j = w, M) \cdot \\ P(Y_i = w|M) \cdot P(Y_j = w|M) \end{array} \right)}{P(d_i|M) \cdot P(d_j|M)}$$

$$= \frac{P(d_i, Y_i = w|M) \cdot P(d_j, Y_j = w|M)}{P(d_i|M) \cdot P(d_j|M)} \quad (12)$$

$$= P(Y_i = w|d_i, M) \cdot P(Y_j = w|d_j, M) \ , \quad (13)$$

which is equal to Equation 1.

As was pointed out above, this derivation embodies a series of assumptions of probabilistic independence. We assume, for example, that the probability of the statistics about different documents d_i are independent of each other, given the information in M. We also assume that the probability of these statistics remains independent, given the additional information that a particular word was drawn from both. We remark that, given the relation we establish in the next section, these assumptions are also present in the use of the cosine coefficient to compute similarity. Our analysis above merely makes these assumptions explicit, opening opportunities for further research on verifying or even finding ways to avoid making them. Such research is, however, beyond the scope of this paper.

2.2 Probability Estimation and Smoothing

We now focus on estimating the term $P(Y = w|d, M)$ in Equation 1.[1] An initial approach is to take the maximum likelihood (ML) estimate for this probability:

$$P_{ML}(Y = w|d, M) = \frac{\xi(w, d)}{\sum_{w \in d} \xi(w, d)} \ , \quad (14)$$

where $\xi(w, d)$ is the number of times that word w appears in document *doc* (represented by the vector d).

This is bound to be a poor estimate, as some words that are "important" to the topic of a document may appear only a few times, whereas other

[1] We drop the subscript previously used with Y for the sake of readability.

"unindicative" terms may appear very often. Also, with shorter documents such as news clips, this estimate will be even more prone to word "spikes" (i.e., will have high variance).

In trying to control variance in estimating $P(Y = w|d, M)$, it becomes critical to perform some type of smoothing. A simple smoothing technique that has been used in the context of computational linguistics [Cha93] is to use the arithmetic mean (AM) of $P_{ML}(Y = w|d, M)$ and the maximum likelihood estimate of the unconditional distribution, $P_{ML}(Y = w|M)$, where

$$P_{ML}(Y = w|M) = \frac{\sum_{doc \in D} \xi(w, d)}{\sum_{doc \in D} \sum_{w \in doc} \xi(w, d)} \ . \tag{15}$$

For the case of $P(Y = w|M)$, the ML estimate is appropriate because this computation is an average over all documents in the entire corpus and is therefore likely to attenuate any word spikes that may appear in a single document. Formally, arithmetic mean smoothing yields

$$\begin{aligned} &P_{AM}(Y = w|d, M) = \\ &\quad \tfrac{1}{2} P_{ML}(Y = w|d, M) + \tfrac{1}{2} P_{ML}(Y = w|M) \ . \end{aligned} \tag{16}$$

Another form of smoothing involves the taking the *geometric* mean (GM) of these two ML distributions[2]:

$$\begin{aligned} &P_{GM}(Y = w|d, M) = \\ &\quad P_{ML}(Y = w|d, M)^{\frac{1}{2}} \cdot P_{ML}(Y = w|M)^{\frac{1}{2}} \ . \end{aligned} \tag{17}$$

The GM estimate in Equation 17 does not define a true probability distribution because it will generally not sum to 1. We thus introduce a true probability distribution based on the geometric mean, by simply adding a normalization factor. This gives us the following *normalized* geometric mean (NGM) estimate:

$$\begin{aligned} &P_{NGM}(Y = w|d, M) = \\ &\qquad \frac{P_{ML}(Y=w|d,M)^{\frac{1}{2}} \cdot P_{ML}(Y=w|M)^{\frac{1}{2}}}{\sum_{w \in W} P_{ML}(Y=w|d,M)^{\frac{1}{2}} \cdot P_{ML}(Y=w|M)^{\frac{1}{2}}} \ . \end{aligned} \tag{18}$$

[2]This is also equivalent to taking the arithmetic mean in the log space: $\log P_{GM}(Y = w|d, M) = \frac{1}{2} \log P_{ML}(Y = w|d, M) + \frac{1}{2} \log P_{ML}(X = w|M)$.

We continue to pursue the unnormalized GM formulation further, since it is related to the computation of similarity between documents using the normalized vector dot product, also known as the "cosine coefficient."

Consider the similarity score of two documents, doc_i and doc_j, computed by using the cosine represented by Equation 19:

$$\sum_{w \in W} \frac{\xi(w, d_i)}{(\sum_{w \in d_i} \xi(w, d_i)^2)^{\frac{1}{2}}} \cdot \frac{\xi(w, d_j)}{(\sum_{w \in d_j} \xi(w, d_j)^2)^{\frac{1}{2}}} \ . \tag{19}$$

The sum in this equation can be reduced to include only those words $w \in d_i \cap d_j$, since any words not in both documents will have $\xi(w, d) = 0$ for at least one of the documents and will not influence the sum. Furthermore, when the cosine similarity score is used in information retrieval and clustering, the raw frequency scores often are not actually used as the features in a document vector. Rather, these frequencies are attenuated by a monotone shrinkage factor such as the log or square root. It has been reported that for the document clustering task, using the square root generally appears to give better performance than using the log [CKPT92]. Incorporating this factor into Equation 19 yields

$$\begin{aligned} &\sum_{w \in d_i \cap d_j} \frac{\xi(w,d_i)^{\frac{1}{2}}}{(\sum_{w \in d_i} (\xi(w,d_i)^{\frac{1}{2}})^2)^{\frac{1}{2}}} \cdot \frac{\xi(w,d_j)^{\frac{1}{2}}}{(\sum_{w \in d_j} (\xi(w,d_j)^{\frac{1}{2}})^2)^{\frac{1}{2}}} \\ &= \sum_{w \in d_i \cap d_j} (\frac{\xi(w,d_i)}{\sum_{w \in d_i} \xi(w,d_i)})^{\frac{1}{2}} \cdot (\frac{\xi(w,d_j)}{\sum_{w \in d_j} \xi(w,d_j)})^{\frac{1}{2}} \ . \end{aligned} \tag{20}$$

Now if we cast Equation 20 in terms of the unnormalized GM estimate defined above, we obtain

$$\sum_{w \in d_i \cap d_j} \frac{P_{GM}(Y_i = w | d_i, M) \cdot P_{GM}(Y_j = w | d_j, M)}{P_{ML}(Y = w | M)} \ . \tag{21}$$

Thus, the cosine similarity metric with square root dampening that has found empirical success in the IR community is actually utilizing a form of geometric smoothing to account for the high variability in word appearances. Furthermore, casting the cosine in our probabilistic framework uncovers a scaling factor for the axes of the word space. Intuitively, this scaling makes sense, since it incorporates additional knowledge in the form of the frequency of word usage in the corpus to be clustered. In our experiments below we

evaluate the expected overlap given by Equation 1 using the various estimation and smoothing proposals introduced in this section, plus a variant that incorporates the scaling factor in Equation 21. As will be seen, the best results are obtained by using the NGM of Equation 18 augmented with the scaling factor from Equation 21 in the denominator.

3 Clustering Algorithms

Having defined a similarity score for documents, we now turn to the problem of the actual document clustering algorithms. While a number of methods for clustering exist, the two most widely applied to text domains are *hierarchical agglomerative clustering* (HAC) and *iterative clustering* techniques such as K-means [Ras92]. Both of these methods rely on the definition of a similarity score between pairs of documents. For the sake of generality, we will refer to this similarity score as $Sim(doc, doc')$ and will subsequently instantiate it with our measure of probabilistic overlap, using different probability estimation methods.

3.1 Hierarchical Agglomerative Clustering

The most common clustering method employed in the information retrieval community over the past decade is HAC [FBY92]. This family of methods begins by placing each document into a distinct cluster. Pairwise similarities between all such clusters are computed, and the two closest clusters are then merged into a new cluster. This process, computing pairwise similarities and merging the closest two clusters, is repeatedly applied, generating a dendogram structure that contains only one cluster (encompassing all the data) at its root. By selecting an appropriate level of granularity in this dendogram, we can generate a partitioning into as many clusters as desired. Criteria such as a minimum number of documents per cluster are often used to prevent outlier documents from being considered a separate cluster. In our experiments we heuristically set this minimum cluster size at 10 documents.

Depending on how the similarity of a document to a cluster is defined,

we can obtain different "flavors" of HAC; the most common are the *single link*, *complete link*, and *group average* methods. Previous work in IR [Wil88] has pointed out that the group average method generally produces superior results. We will concentrate on this method in this paper.

The group average method defines the similarity between a document *doc* and a cluster C as the average of the pairwise similarities between *doc* and each of the documents in C: $Sim(doc, C) = \sum_{doc' \in C} \frac{1}{|C|} Sim(doc, doc')$.

A simple probabilistic interpretation of the group average method is that each document in a cluster is an equally likely representative of that cluster. This is evident in the $\frac{1}{|C|}$ weighting given to each term in the sum. Note that we can obtain many variations of HAC by replacing the term $\frac{1}{|C|}$ with alternate distributions over the "weight" of documents in a cluster (e.g., a Gaussian based on a document's distance from the cluster centroid).

3.2 Iterative Clustering

Iterative clustering techniques, also referred to as *reallocation* methods, attempt to optimize a given clustering by repeatedly reassigning documents to the cluster to which they are most similar. The general form for such algorithms, given a specification of the number of clusters k, is as follows.

1. Initialize the k clusters.[3]
2. For each document *doc*, compute the similarity of *doc* to each cluster.
3. Assign each document *doc* to the cluster to which it is most similar.
4. Goto 2, unless some convergence criterion is satisfied.

As in the case of HAC, we define the similarity of a document to a cluster by the group average similarity. Our exit criterion in Step 4 can be met by simply running the algorithm for 10 iterations (although we observed that often far fewer were needed for convergence.)

We note that the initialization in Step 1 will affect the convergence point of the algorithm. We experimented using various runs with random initial

[3]The random assignment of documents to clusters is one simple method of initialization.

Data Set	Number of Documents	Number of Words	Categories	Baseline Error Rate
D1	486	1143	nat-gas, soybean, dlr	53.3%
D2	466	1001	gold, coffee, sugar	59.0%
D3	289	552	tbill, yen, reserves	56.1%
D4	467	1126	gnp, livestock, sugar	60.4%
D5	1426	1953	loan, interest, money-fx	57.1%

Table 1: Data sets used in clustering experiments.

Data Set	U-ML	U-AM	U-NGM	S-ML	S-AM	S-NGM	Cos	TFIDF
D1	0.14	0.09	0.22	0.27	0.30	0.34	0.41	0.26
D2	0.16	0.11	0.26	0.35	0.38	0.43	0.47	0.28
D3	0.25	0.22	0.42	0.35	0.42	0.49	0.54	0.35
D4	0.17	0.11	0.31	0.34	0.38	0.48	0.52	0.32
D5	0.26	0.16	0.40	0.37	0.41	0.48	0.63	0.47

Table 2: Ratios of average between label to within label similarity.

clusters, and with using HAC as a method to find an initial clustering. The results of the former were often comparable and in some cases worse than the later. For reasons of space we report only on the experiments where HAC determined the initial set of clusters.

4 Results

The objective of the experiments we describe in this section is to test the different estimation schemes for the computation of the expected overlap between documents. We are also interested in evaluating the effect of axis scaling on the expected overlap measure revealed from the derivation of the cosine coefficient. As will be seen below, the scaled NGM score of overlap performs better (in some case dramatically better) than any other score we

tested, including the cosine coefficient and TFIDF weighting method commonly used in IR [SB87].

Realizing that methods for evaluating clustering algorithms are not without controversy, we use the following strategy (keeping aware of its limitations). We use previously labeled data and measure how well the clustering recovers the known label structure in the data. To this end, we specified the number of clusters to be the number of known class labels in the data. The clustering algorithm, however, is given no information about the true label of each document. After clustering is completed, we designate the predicted label for all documents in each cluster to be the true label that the majority of documents in that cluster have. Once we have an actual and predicted label for each document, we can now simply compute the classification error. This also gives us a baseline (maximal) error for each data set, which we would get if all instances were classified in the majority class.

Our experiments were conducted on various subsets of the Reuters-22173 data set.[4] We expect that the results of selecting a corpus such as the Reuters-22173 news articles will be that the labeling will indeed reflect some semantic coherence that can be trusted for evaluation. The data sets used here were created from only documents with one of a particular subset of class labels from the Reuters collection. We also applied a simple preprocessing feature selection to these data sets using a standard Zipf's Law analysis to eliminate any words that appeared fewer than 10 or greater than 1000 times, as providing too little discriminating power between documents. A description of these data sets is given in Table 1.

Seeking to characterize the data sets in our study according to the difficulty of recovering the underlying class structure, we also measured the ratio of the average interlabel similarity with the average intralabel similarity. These values, shown in Table 2, indicate the relative difficulty we would expect each measure of similarity to have with each data set. An increase in these values indicates that documents within a class appear more and more similar to documents outside the class, thus making the recovery of the true class structure much more difficult. From these values we find that data sets D1, D2, and D3 are clearly in order of increasing difficulty for *all* the simi-

[4] An updated version of this data set, Reuters-21578, is now publically available from David Lewis at `http://www.research.att.com/~lewis`.

Data Set	U-ML	U-AM	U-NGM	S-ML	S-AM	S-NGM	Cos	TFIDF
D1	4.3%	4.3%	4.9%	5.3%	4.7%	2.3%	1.0%	3.9%
D2	6.9%	4.1%	1.1%	13.3%	9.0%	4.7%	5.4%	9.4%
D3	31.8%	22.5%	23.2%	11.8%	16.3%	3.1%	20.8%	12.5%
D4	24.0%	47.3%	23.8%	9.4%	10.5%	5.4%	55.9%	42.4%
D5	25.8%	21.5%	55.8%	35.2%	27.2%	26.5%	50.3%	50.8%

Table 3: Error rates from hierarchical agglomerative clustering.

Data Set	U-ML	U-AM	U-NGM	S-ML	S-AM	S-NGM	Cos	TFIDF
D1	2.1%	1.0%	1.9%	0.8%	0.8%	0.6%	0.6%	0.6%
D2	2.6%	0.9%	1.5%	7.5%	3.0%	1.5%	3.0%	4.3%
D3	31.5%	20.4%	20.4%	4.8%	5.5%	4.2%	23.3%	7.6%
D4	10.0%	9.5%	11.9%	6.6%	6.6%	4.1%	19.1%	40.0%
D5	24.7%	32.0%	31.6%	29.2%	24.9%	20.3%	22.4%	47.3%

Table 4: Error rates from iterative clustering using HAC seeding.

larity measures. Data sets D4 and D5 show more relative variability, which is reflected in the results of our experiments.

We empirically evaluated our measure of probabilistic overlap, using a number of different estimation schemes. First, we computed document overlap using the ML, AM, and GM estimates for $P(Y = w|d, M)$. In these cases we did not scale the axes of the word space, so these computations are denoted "Unscaled" (U-). We then modified the computation of document overlap to include a scaling factor based on the marginal probability of word appearance, yielding

$$\sum_{w \in d_i \cap d_j} \frac{P(Y_i = w|d_i, M) \cdot P(Y_j = w|d_j, M)}{P_{ML}(Y = w|M)} \; . \tag{22}$$

We identify these runs as "Scaled" (S-). For comparison, we also performed clustering using the cosine coefficient (with square root dampening) as a similarity score as in Equation 21. Also, recognizing the use of TFIDF weighting

in the IR literature [SB87] as an alternate means of term scaling, we also used this weighting scheme, in conjunction with the Cosine rule (without square root dampening), as yet another similarity score for comparison. For our TFIDF weighting we used the commonly used scheme: $TF(w,d) = \xi(w,d)$ and $IDF(w) = \log(\frac{N}{n_w})$, where N is the total number of documents and n_w is the number of documents in which word w appears at least once.

The error rates for clustering using HAC are given in Table 3. Those for iterative clustering, using HAC as an initialization, are given in Table 4. We note that applying the iterative optimization after performing HAC almost always leads to improved results, as seen in the reduction in error rates from Table 3 to Table 4. Hence, we focus our attention on Table 4.

Our first conclusion is that the use of axis scaling often improves the performance of the similarity measure using ML, AM, and NGM estimates. As a matter of fact, the error rate is reduced in 11 cases (often drastically), is increased in 3 cases (only slightly), and remains unchanged in 1 case. To investigate whether a measurable characteristic in the data sets themselves points to the benefit of using scaling, we performed a chi-squared test on each data set. The purpose of this test is to check the hypothesis that the marginal probabilities of each word in a data set are uniformly distributed, in which case we would expect scaling not to help. As could be expected, the hypothesis of uniformity was rejected for every data set with an error probability of less than 10^{-6}.

Our second, and most important, conclusion highlights the utility of S-NGM as a similarity score. In general, the scaled probabilistic similarity measures using ML, AM, and NGM perform extremely well in comparison to both the cosine and TFIDF similarity scores, which are currently the state of the art in information retrieval. Most significantly, we draw the reader's attention to the S-NGM similarity score, which *always* produces an error rate comparable to or significantly less than that of either the cosine or the TFIDF methods! Noting that the cosine coefficient is equivalent to a scaled but unnormalized GM estimate, we see that the use of normalization to obtain true probabilities, as in the S-NGM case, not only can preserve the clean, well understood probabilistic semantics of our overlap measure, but also can have a significant beneficial impact on the empirical performance.

5 Alternative Probabilistic Models

An alternative approach to text clustering is based on the use of probabilistic mixture modeling, such as the AutoClass system [CKS+88]. In our investigations of this approach, documents were represented as binary vectors (rather than word frequency counts). AutoClass was used to cluster documents as mixtures of independent binomial distributions over word appearances. This representation has two immediate consequences: (1) it loses word frequency information and (2) it treats evidence about whether or not a word appears in a document in a symmetrical manner.

The loss of word frequency estimation may be remedied by the use of more complex statistical models (e.g., parametric distributions, such as Gaussians or Poissons, over word frequencies) to fit the data. This approach, however, requires a commitment to a particular parametric model of word appearance. Our initial investigation along these lines, using Gaussian distributions, indicates that this approach may not be promising.

In the context of text clustering, the symmetrical treatment of evidence is more problematic. By "symmetrical treatment" we mean that word appearance and absence are given the same weight in a binomial distribution such as the one described above. One would expect, however, that the appearance of particular words in a text would be more indicative of a particular topic than the absence of some other word. Note that our probabilistic model (which is based on a single multinomial) proposed in the overlap score places much more importance on the information about the appearance of words than on their absence. Thus, the model matches our intuitions about word usage in text.

To test these arguments, we convert the data sets previously described to binary representations. The objective is to compare the two probabilistic models on fair grounds by removing the word frequency information. We then cluster this data, using the S-NGM and cosine similarity scores. As before, we use both HAC alone and HAC followed by iterative clustering as the clustering methods. We also run AutoClass (which is a priori given the proper number of clusters to find), with initial clusters set with the results from HAC or randomly. To help alleviate the problems with bad initial conditions in the random case, we run AutoClass multiple times with different random initial clusters, and report the results for the best clustering chosen according to AutoClass's own model selection criterion. The results

	HAC		HAC + Iter		AutoClass		
Data Set	S-NGM	Cos	S-NGM	Cos	S-NGM	Cos	Random
D1	2.3%	39.5%	0.4%	0.4%	2.1%	38.3%	53.3%
D2	7.7%	35.8%	5.4%	13.9%	9.0%	35.6%	54.3%
D3	29.1%	22.8%	29.1%	15.6%	35.3%	27.3%	47.4%
D4	24.0%	51.8%	14.6%	43.3%	29.4%	53.6%	46.4%
D5	22.6%	50.8%	22.7%	43.8%	27.8%	51.6%	40.3%

Table 5: Error rates using binary data.

of these experiments are given in Table 5.

As expected, the lack of word frequency information generally hinders both S-NGM and cosine across both non-AutoClass clustering regimes. Most striking, however, is the poor performance of AutoClass on any of the text data sets. AutoClass with random initialization fails to find any real structure in any of the data sets. Furthermore, even when "reasonable" initial clusters are provided by HAC (using S-NGM and cosine), AutoClass outputs final clusters that are much worse than the other methods.

As an aside, we note that while this intuition about asymmetry of evidence is useful for text clustering where categories must be discovered, it generally does not hold true for text classification tasks where the categories are known a priori. This has also been observed empirically in the successful application of such symmetric probabilistic models to classification problems in text [LR94] [KS97] and other other domains [FGG97]. A full discussion of this point is beyond the scope of this paper.

6 Conclusion

We have presented a probability-based measure for document similarity that is quite effective for clustering. We have also shown how the widely used cosine similarity coefficient can be captured as a particular form of probability estimation within our framework. Moreover, this formulation of the cosine coefficient has revealed a scaling factor that can be effectively integrated

into our probabilistic framework and that yields results superior to those of traditional IR methods.

In future work we will seek to extend our probabilistic similarity score to include arbitrary functions over words in documents (such as phrases and logical operations). This can be done by expanding the domain of the multinomial distributions we currently use to compute expected document overlap. In this way we will be able to easily incorporate much more information than word frequencies into our similarity score. We also wish to extend the use of such a similarity measure to problems in other domains such as video segmentation, using different estimation techniques as appropriate. We have obtained promising initial results on such problems.

As a long-term goal, we plan to use the well understood probabilistic semantics of our model as leverage in developing a clean fusion of information from different modalities to aid in multimedia information retrieval.

References

[Cha93] E. Charniak. *Statistical Language Learning.* MIT Press, Cambridge, MA, 1993.

[CKPT92] D. R. Cutting, D. R. Karger, J. O. Pederson, and J. W. Tukey. Scatter/Gather: a cluster-based approach to browsing large document collections. In *Proceedings of ACM/SIGIR*, pages 318–329, 1992.

[CKS+88] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman. AutoClass: a Bayesian classification system. In *Proceedings of Machine Learning*, pages 54–64, 1988.

[FBY92] William B. Frakes and Ricardo Baeza-Yates. *Information Retrieval: Data Structures and Algorithms.* Prentice Hall, 1992.

[FGG97] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[Fuh89] Norbert Fuhr. Models for retrieval with probabilistic indexing. *Information Processing and Management*, 25(1):55–72, 1989.

[HP96] Marti A. Hearst and Jan O. Pederson. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of ACM/SIGIR*, 1996.

[KS97] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In *Proceedings of Machine Learning*, pages 170–178, 1997.

[LR94] David D. Lewis and M. Ringuette. Comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR*, 1994.

[PSHD96] Peter Pirolli, Patricia Schank, Marti Hearst, and Christine Diehl. Scatter/gather browsing communicates the topic structure of a very large text collection. In *Proceedings of CHI*, 1996.

[Ras92] E. Rasmussen. Clustering algorithms. In William B. Frakes and Ricardo Baeza-Yates, editors, *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, 1992.

[Sal71] G. Salton. *The SMART Information Retrieval System*. Prentice Hall, Englewood Cliffs, NJ, 1971.

[SB87] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical Report 87-881, Cornell University Computer Science Department, November 1987.

[SS97] Hinrich Schuetze and Craig Silverstein. A comparison of projections for efficient document clustering. In *Proceedings of ACM/SIGIR*, 1997.

[vR79] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[vRJ71] C. J. van Rijsbergen and N. Jardine. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7:217–240, 1971.

[Wil88] Peter Willett. Recent trends in hierarchical document clustering: A critical review. *Information Processing and Management*, 24(5):577–597, 1988.