An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering

Vasileios Hatzivassiloglou Luis Gravano Ankineedu Maganti

Department of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027, USA
{vh, gravano, amaganti}@cs.columbia.edu

Abstract

We investigate four hierarchical clustering methods (single-link, complete-link, groupwise-average, and single-pass) and two linguistically motivated text features (noun phrase heads and proper names) in the context of document clustering. A statistical model for combining similarity information from multiple sources is described and applied to DARPA's Topic Detection and Tracking phase 2 (TDT2) data. This model, based on log-linear regression, alleviates the need for extensive search in order to determine optimal weights for combining input features. Through an extensive series of experiments with more than 40,000 documents from multiple news sources and modalities, we establish that both the choice of clustering algorithm and the introduction of the additional features have an impact on clustering performance. We apply our optimal combination of features to the TDT2 test data, obtaining partitions of the documents that compare favorably with the results obtained by participants in the official TDT2 competition.

1 Introduction

Clustering plays a crucial role in organizing large document collections. As a notable example, clustering can be used to structure query results, hence providing users with an overview of the results that is easier to understand and process than a flat list of documents (see, e.g., [7]). It can also form the basis for further processing of the documents once they have been organized in topical groups, such as summarization [11].

Clustering is also a key component of DARPA's ongoing Topic Detection and Tracking (TDT) initiative, which completed its second phase (TDT2) in early 1999.[1] The goal of the TDT initiative is to provide benchmarks for comparing systems that address three specific tasks relating to manipulating and organizing broadcast news and newswire stories. Given a stream of incoming news articles, *topic detection* is the task of grouping together the articles that correspond to the same *topic*, where a topic is defined as "a seminal event or activity, along with all directly related events and activities." [4, p. 19]. Topic detection is then a clustering task, where we group documents on the same "topic" together.

In this paper, we study document clustering applied to the TDT2 topic detection problem. For this, we investigate alternatives for the two crucial components of a clustering strategy, namely the clustering algorithm itself and the document features that are used to guide the clustering. More specifically, we study the performance of the four most popular hierarchical clustering algorithms, single-link, complete-link, groupwise-average, and single-pass clustering. (We do not consider in our evaluation more expensive, non-hierarchical clustering techniques because of efficiency concerns.) Single-pass clustering makes irrevocable clustering assignments for a document as soon as the document is first inspected. Among the four techniques that we considered, single-pass is then the best suited for the TDT2 topic detection task, which requires systems to make clustering assignments "on-line" as soon as a new document is received. To investigate the limitations of such an on-line algorithm, we experimentally compare the performance of single-pass with the other three clustering algorithms mentioned above.

The other component of a clustering strategy that we explore in this paper is the document features that guide the clustering. Typically, document clustering techniques use the words that appear in the documents to define the "distance" function that determines the final clustering. But additional, more linguistically informed sets of features can be used in an attempt to limit the input features to the most important ones, facilitating the task of the learning (i.e., clustering) algorithm. In this paper we investigate two such sets of automatically identified features: matched noun phrase heads, where additional premodifiers are excluded, and proper names (single nouns and phrases), categorized as people, place, or organizations' names.

We conduct a large-scale experimental evaluation involving over 40,000 real documents from the TDT2 initiative 1998 data set. Our results show that, as expected, groupwise average outperforms the other hierarchical clustering algorithms, but (for a limited range of clustering thresholds) its performance is surprisingly close to that of the computationally cheaper, on-line

[1] See http://www.itl.nist.gov/iaui/894.01/tdt98/tdt98.htm.

single-pass method. We also establish that the linguistically motivated features increase the overall clustering performance when used in conjunction with the full word vectors traditionally employed in clustering. Our results compare favorably with those obtained by the official participants in the TDT2 competition.

In the remainder of the paper, we first review the four clustering algorithms and linguistic features we experimented with (Sections 2 and 3, respectively). Since combining the features is an important step that is usually approached with approximate and computationally expensive exhaustive search methods, we present in Section 4 a statistical model for this combination task based on a trainable log-linear model. Section 5 contains a detailed presentation of our experimental results, along with a discussion of their significance.

2 Clustering Algorithms

Part of the goal of the experiments reported in this paper is to explore the effect that the chosen clustering algorithm has on the quality of document clustering (i.e., clustering with text features). To this end, we implemented the four major hierarchical clustering technique discussed in the literature. These techniques (or their variants) are used by most IR systems that perform clustering of thousands of documents. We briefly review the four algorithms below, and discuss their strengths and weaknesses. (See [5] for a more complete discussion of hierarchical clustering techniques.)

Given a set of documents $\mathcal{S} = \{D_1, D_2, \ldots, D_n\}$ and a similarity function $f : \mathcal{S} \times \mathcal{S} \rightarrow \Re$, the goal of all four methods is to output a "reasonable" partition of $\mathcal{S}$ into a collection of clusters $C_1, C_2, \ldots, C_m$ as a function of a pre-selected threshold τ. The first three techniques (*single-link clustering*, *complete-link clustering*, and *groupwise-average clustering*) share the same algorithmic steps but differ in the criterion they apply to determine when two clusters should be merged. The general procedure first places each document D_i in a separate cluster C_i (i.e., initially $m = n$); then it iteratively examines pairs of clusters, merges a pair of clusters that satisfy the method's test for merging, and repeats this cycle until no mergeable clusters can be found, reducing m by one at each iteration. The merging criterion is what distinguishes each of these three methods:

- For single-link clustering, two clusters C_i and C_j can be merged if there is a pair of documents $x \in C_i$ and $y \in C_j$ such that $f(x, y) \geq \tau$.
- For complete-link clustering, two clusters C_i and C_j can be merged if all pairs $x \in C_i$ and $y \in C_j$ satisfy $f(x, y) \geq \tau$.
- Groupwise average adopts a middle position between these two extremes. Two clusters C_i and C_j are mergeable if the average similarity $f(x, y)$ across all pairs with $x \in C_i$ and $y \in C_j$ is equal to or greater than the threshold τ.

The produced clustering also depends on which pair of clusters is actually merged at each iteration, whenever there are multiple candidates. Single-link clustering is actually unaffected by this selection, since if C_i and C_j are mergeable at some point, they will continue to be mergeable even if additional elements are included in one or both of these clusters. For complete link and groupwise average, we chose (as most other systems do) to select for merging the pair of clusters with the largest minimum or average similarity, respectively.

The fourth technique we experimented with, *single-pass clustering*, is unique in that it makes (irrevocable) clustering assignments as soon as it sees each element.[2] This makes single-pass clustering especially suitable for very large collections of documents, and also for situations where new documents arrive continually, in different points in time (e.g., as is the case with an online news source). The algorithm proceeds by maintaining an initially empty but ever increasing set of clusters. As each new document arrives, its average similarity with all the members of each existing cluster is calculated. If a cluster for which this average exceeds or matches the threshold τ is found, we assign the document to that cluster. Otherwise, a new cluster containing only the just arrived document is formed. Again, we choose among multiple clusters that satisfy the similarity criterion by selecting the one that has maximum average similarity with the document under consideration.

Among these four methods, single- and complete-link reduce the numerical similarity information to the minimum or maximum of a set. Therefore, these methods are expected to perform less well (and run faster) than the groupwise-average technique, which takes into account information across all pairs in the clusters it assesses for merging. In comparison, single-pass clustering operates with less information at each step, since each document must be placed in its final cluster as it arrives. Hence, we anticipate that single-pass clustering will underperform groupwise average.

All these methods adopt a greedy approach to clustering, which is justified for very large data sets. An alternative class of algorithms, collectively referred to as *non-hierarchical* clustering methods, spend much more time per clustered element in order to improve the quality of the partitioning. They do so by defining a merit function for the entire partition, which is then iteratively optimized; unlike the four hierarchical techniques discussed above, they can revise prior decisions, moving elements out of the cluster they had originally been assigned. This is the reason for both the increased performance and the increased computational complexity of these techniques. See [8] for a comprehensive discussion of non-hierarchical techniques, including the k-means, k-medians, exchange, and simulated annealing methods. Since in the experiments reported in this paper we worked with tens of thousands of documents, collections that even hierarchical methods take hours to cluster, we did not include optimization methods in our comparative analysis.

3 Linguistic Features

Document clustering is a task that has received considerable attention in the IR community, and the recent and ongoing Topic Detection and Tracking effort has highlighted the issues involved when very large collections of documents are partitioned. Many techniques for improving the basic algorithms described in the previous section have been considered, including multiple clustering stages with varying thresholds [22] and probabilistic

[2] Delaying these decisions for a fixed number of elements is possible, but we did not include such a delay in our implementation.

mixture models of word vector distributions [10]. Yet, any learning method depends on the selection of the most informative input features for producing high quality output. In the TDT effort, and in most other clustering work for information indexing and retrieval purposes, the words in the document have dominated as the sole features on which the clustering is based.

A good case can be made for the suitability of the full collection of words as the basis for determining document similarity and eventually clusters. After all, the unfiltered words contain all the information that humans have when they perform the same task. Keeping all those words, appropriately weighed with a scheme such as `tf.idf` [15], makes all this information available to the clustering algorithm, and avoids hard choices such as limiting the similarity features to specific words or classes of words. However, for many tasks, an informed selection of features can prove beneficial by injecting external, linguistic knowledge about what kinds of words are most important for the classification, thus enabling the learning algorithm to zoom in to the most significant input features. For example, in work classifying images on the basis of their captions, Sable and Hatzivassiloglou [14] have established that keeping only the first sentence of image captions and only specific parts of speech significantly improves classification accuracy. Other types of linguistic knowledge have also been found useful for information retrieval tasks (e.g., nominal compounds [6], syntactic constraints [16], and collocations [18]).

In this paper, we explore two linguistically motivated restrictions on the set of words used for clustering: noun phrase heads and proper names. While it is not possible to predetermine with complete accuracy which word classes make a document belong to a specific topical class, noun phrases and proper nouns carry most of the information about the protagonists in each document, and (indirectly) about the location and time frame of any events discussed therein. This assumption is more justified for news articles, where 80–85% of documents describe one or more specific *events*, as opposed to generic discussions of a topic [9]. By limiting the input features to those grammatical categories, we construct additional feature vectors that can be used either in place of or in addition to the traditional word vectors during clustering.

We use two external tools to extract these features automatically from text. For identifying noun phrase heads, we use *LinkIt* [19], a tool developed at Columbia University for the purpose of identifying significant topics in documents and indexing text collections. LinkIt uses part-of-speech information (also automatically assigned using the Alembic toolkit developed at MITRE [1]) and a simple finite-state grammar to locate maximal non-recursive noun phrases in the text. Then it collates those phrases that have the same head (final noun in the sequence). In this manner, the phrases "Bill Clinton", "President Clinton", and "Clinton" will all be mapped to *Clinton*, providing a means for addressing the hard problem of definite coreference. Unfortunately, so will "Hillary Clinton", demonstrating that the approach will also introduce some reference errors. Our experiments measure whether the positive contribution of collapsing related terms to a canonical form with LinkIt's basic implementation outweighs these errors.

The second tool we use, *Nominator* [20], was developed at IBM. It uses capitalization and punctuation information together with a contextual model and a large knowledge base to identify proper nouns in context. Not only are proper names recognized, but they are also classified into categories such as PERSON, PLACE, and ORG (the latter standing for "organization"). This allows us to experiment with different versions of proper name vectors for each document, by including different categories in our definition of what really is a proper name. We considered three such versions:[3]

- Our first version (hereafter referred to as *NominatorAll*) simply takes all the words or phrases labeled as proper names by Nominator.
- For the second version (*NominatorPPOU*), we exclude all words and phrases labeled as OTHER or UTERM (unknown term). These are words or phrases that Nominator is either unable to confidently characterize as proper names, or identifies as fixed terms, respectively; in both cases the probability of their being a proper name is lower. Examples of words tagged as OTHER in the TDT2 corpus include "Internet" and "Chapter 7", while examples of UTERM are "private school" and "recent study".
- Finally, we only consider words and phrases marked as PERSON, PLACE, or ORG (version *NominatorPPO*). These are the classes most likely to include information about the participants and location of an event, which we consider central to a document's placement in an appropriate cluster.

4 Combining Features

The previous section presented ways to calculate vectors of linguistically motivated features, which can complement the basic vector containing all words that appear in a document. Having extracted those vectors, we can then calculate the similarity between documents by first applying a `tf.idf` normalization on each vector and then taking the cosine between the two vectors that correspond to the two documents. In this manner, we obtain different similarity matrices (or functions), depending on which set of input features is chosen. These similarity matrices can then be used individually to drive the clustering algorithm and produce a partition of the document set. Nevertheless, it is interesting to also consider combinations of the similarity values assigned to a pair of documents by the different feature models.

How to combine such similarity values is a hard problem that involves two steps: first, deciding the form that the combining function should have, and second, assigning values to the parameters in that form. Frequently, the chosen form is a linear weighted sum, and the weights are estimated via a search regime that calculates the final similarity for a given set of weights, clusters the documents, and evaluates the produced clustering, repeating the process iteratively for different sets of weights. The complexity of this approach is exponential in the number of weights, and consequently it cannot be used with more than a few such parameters. Even then, the exhaustive search is limited in the range and resolution of the weights considered, and often has to be approximated by either gradient-descent or decomposition techniques.

[3]In all three versions, we distinguish between occurrences of the same word that are assigned different labels in different contexts, e.g., *Ford*/PERSON and *Ford*/ORG.

To avoid these problems, we consider a mathematical model for selecting optimal values for the weights in a slightly modified formulation of the weighted sum approach. Given vectors of similarity values $\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_k$, one for each feature (*Words*, *LinkIt*, *NominatorPPO*, etc.) and with each value V_{ij} corresponding to the j-th pair of documents, we assume that the combining function should best approximate the values of a vector $\boldsymbol{R}$ (again ranging over the same pairs of documents), where

$$\boldsymbol{R}_j = \begin{cases} 1 & \text{if the documents in the } j\text{-th pair should be in the same cluster} \\ 0 & \text{otherwise} \end{cases}$$

The values $\boldsymbol{R}_j$ can be obtained from a training set of documents for which optimal cluster assignments are available (this is the case in the TDT2 training data used in our experiments). Then, we fit a *log-linear regression model* [17] in which the $\boldsymbol{V}_i$'s are the predictors and $\boldsymbol{R}$ the response. Such a model calculates first a linear internal predictor, $\boldsymbol{\eta}$, which is a weighted sum of the $\boldsymbol{V}_i$'s,

$$\boldsymbol{\eta} = \sum_{i=1}^{k} w_i \cdot \boldsymbol{V}_i$$

and then relates $\boldsymbol{\eta}$ to the final response via the *logistic transformation*

$$\boldsymbol{R}_j = \frac{e^{\boldsymbol{\eta}_j}}{1 + e^{\boldsymbol{\eta}_j}}$$

Note that the log-linear model is quite similar to linear regression, which has been used successfully for combining features for text classification before [21]. The log-linear extension guarantees that the final response will lie in the interval $(0, 1)$, with each of the endpoints associated with one of the outcomes. It is also more appropriate than a straightforward weighted sum because of technical reasons relating to the statistical assumptions inherent in such modeling (the fact that the variance in the binomial distribution, which appropriately models each $\boldsymbol{R}_j$, is dependent on the mean and not constant as assumed by the linear regression model; see [17]). Given very modest assumptions about the distribution of the $\boldsymbol{V}_i$'s, the optimal set of weights w_i can be calculated efficiently using the *iterative reweighted nonlinear least squares* algorithm [2].

This approach aims to optimize the final similarity function, rather than the evaluation metrics over the clustering that this function leads to. Hence, it is possible that a different set of weights would lead to higher overall scores, when the effects from the particular clustering algorithm are factored in. However, in our experimental analysis (Section 5) we found that the weights assigned by this procedure were in all but one case better than those produced by (limited) exhaustive search; and in the one exceptional case, the difference in final performance was negligible. At the same time, the log-linear model greatly simplifies the task of combining the similarity values in a principled manner, and allows us to experiment with more models (e.g., *Words* plus *LinkIt* data alone, *Words* plus each of the versions of Nominator data alone, etc.) than would otherwise be practical.

5 Experimental Evaluation

In the previous sections we described the possible choices of clustering algorithms and of text features that we can use to produce document clusters. We now evaluate the clustering and feature selection choices experimentally (Section 5.2), using text corpora and evaluation metrics developed in the context of the TDT2 initiative (Section 5.1). In addition to reporting results for the single-pass algorithm, which satisfies TDT2's online requirement, we also present results for the other three clustering algorithms of Section 2, even though these three algorithms do not fit strictly in the TDT2 setting, since they inspect the documents repeatedly during clustering.

5.1 TDT2: Corpora and Metrics

The TDT2 corpus that we use for our experiments consists of newswire articles from 1998, from Associated Press, The New York Times, the Voice of America, Public Radio International, CNN, and ABC. Our training corpus consisted of the 20,228 articles in the TDT2 training corpus, while our evaluation test corpus consisted of the 22,410 documents in the TDT2 evaluation test corpus for topic detection.[4]

The TDT2 metric for evaluating the performance of topic detection systems is the *cost of detection*, `cdet`, which combines miss (P_M) and false alarm (P_{FA}) errors into a single number,

$$\texttt{cdet} = C_M \cdot P_M \cdot P_T + C_{FA} \cdot P_{FA} \cdot (1 - P_T)$$

where C_M and C_{FA} are the costs of a miss and a false alarm, respectively (equal to 1 in the TDT2 evaluation), and P_T is a training set specific a priori target probability of a story discussing a topic (equal to 0.02 for the TDT2 evaluation of topic detection). Note that P_M and P_{FA} are related to the more traditional metrics of recall and precision, respectively; more specifically, P_M is equal to 1 minus recall, and P_{FA} is the same as fallout (which generally is low when precision is high). `cdet` offers one way to combine these usually competing factors into one number, so that rankings of different systems, or different versions of the same system with different input features, can be made.

The results that we report next use the `cdet` metric above, and were computed with evaluation software produced by NIST for TDT2. Using the same training and test corpora[5] and the same evaluation metric enables us to directly compare results with those obtained by the TDT2 participants in a large-scale evaluation. [12] discusses further details of the TDT2 evaluation methodology, including the way that system-produced clusters are aligned with the clusters in the reference model. In both TDT2 and our experiments presented here, evaluation scores are reported separately after micro-averaging (i.e., scores are calculated per document and averaged across all documents) and macro-averaging (i.e., scores are calculated per cluster in the reference model and averaged across those clusters), or, in TDT terminology, as *story-weighted* and *topic-weighted*, respectively.

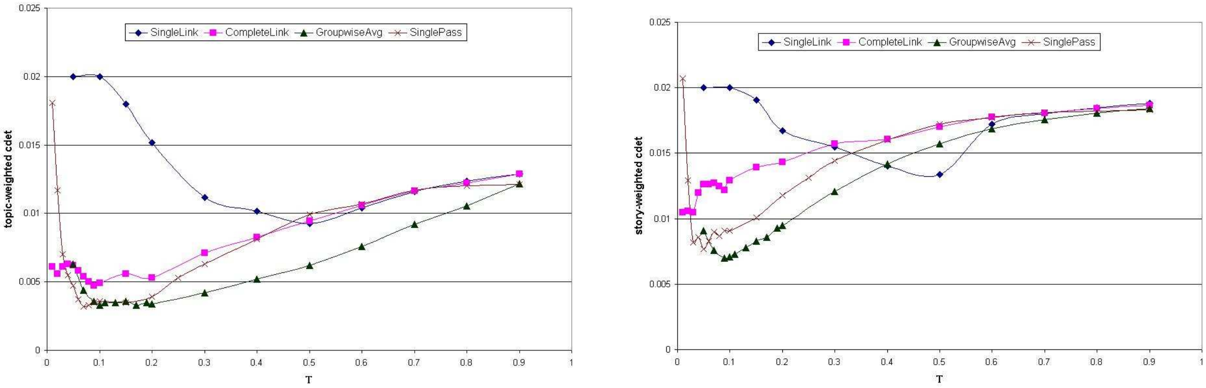
5.2 Results over the Training Corpus

`Tf` vs. `tf.idf` weights:

As a first experiment, we studied the performance of the `tf` and `tf.idf` weighting schemes over the TDT2 training cor-

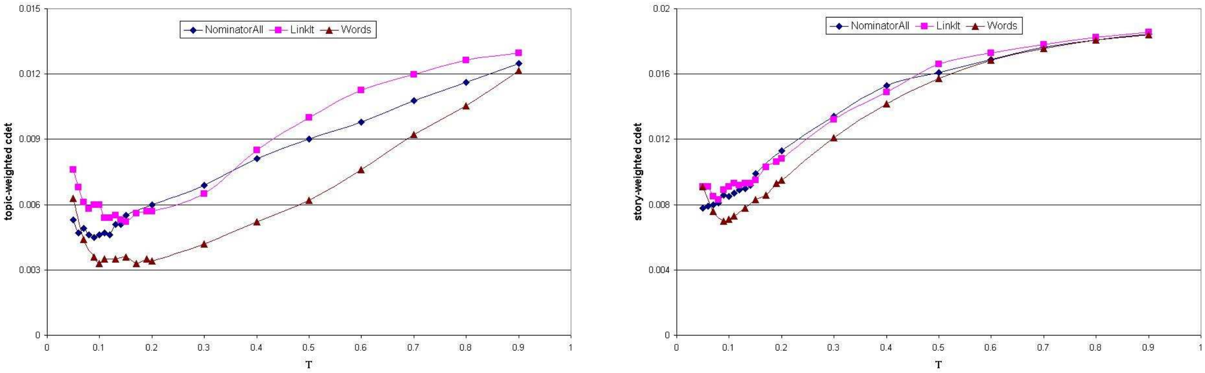
[4] The TDT2 data included 21,950 additional documents in a *development* set that participants were free to use during training. We did not use these documents for the experiments reported in this paper.

[5] Actually, a subset of the training corpus available to TDT2 participants; see footnote 4.



(a) **(b)**

Figure 1: Topic- **(a)** and story-weighted **(b)** `cdet` as a function of the threshold τ (*Words* with `tf.idf` weights).



(a) **(b)**

Figure 2: Topic- **(a)** and story-weighted **(b)** `cdet` as a function of the threshold τ (`tf.idf` weights; groupwise-average clustering).

pus discussed above. Not surprisingly, using `tf.idf` weights results in slightly better (i.e., lower) values of both the story- and the topic-weighted `cdet` metric. Hence, in the rest of our experimental evaluation we use `tf.idf` weights. Also, in all subsequent experiments we adopted a standard cosine metric to calculate the similarity between two feature vectors.

Choice of clustering algorithm:

Figure 1 compares the performance of the clustering algorithms that we discussed in Section 2, for the *Words* representation of the documents using `tf.idf` weights. As it was expected on theoretical grounds, groupwise-average performs significantly better than both complete-link and single-link, and better than single-pass clustering. These results are consistent across the other feature representations of the documents that we tried (Section 3). Consequently, we mostly focus on groupwise-average clustering in the rest of the paper. However, we also report results for single-pass clustering. In effect, although the latter technique performs slightly worse than groupwise-average clustering over our training data, it has the important advantage of being an on-line technique (Section 2), comparable to the methods used in TDT2. Furthermore, Figure 1 reveals that for an appropriate range of thresholds τ, single-pass performs almost as well as groupwise-average clustering. This surprising fact justifies the use of single-pass clustering if accurate estimation of a suitable value of τ can be performed from the training data, and may offer an explanation for the relatively small improvement obtained by TDT2 systems that delayed clustering decisions for 10 or 100 documents.

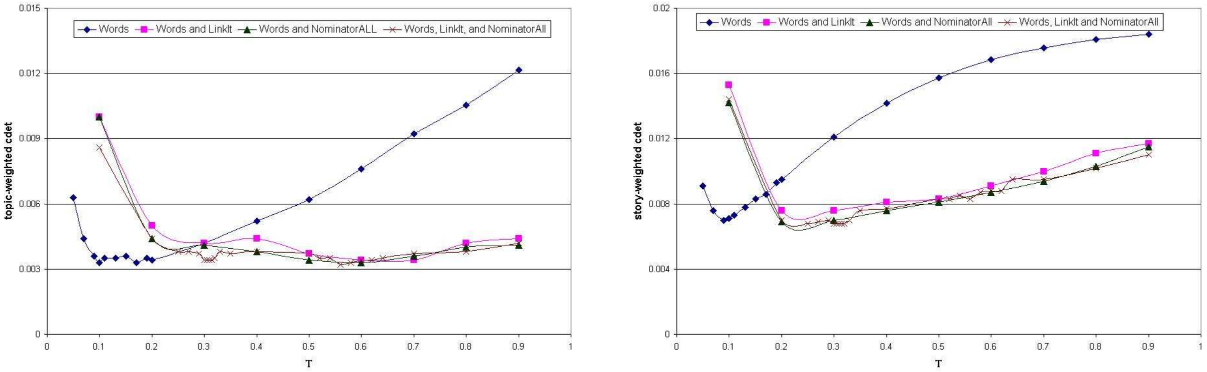
Analysis of individual document features:

As discussed in Section 3, we have a choice of features that we can use to represent the documents in our collection. Figure 2 shows the `cdet` values that we obtain when we use each of these choices in isolation for clustering. As we can see, the *Words* representation performs the best, with significantly lower values of both topic- and story-weighted `cdet` than those for the *LinkIt* and *NominatorAll* representations.

This result may indicate that discarding non-nouns results in significant loss of information, or may be due to limitations of the specific tools (LinkIt and Nominator) we used. For example, many cases of slightly different forms of proper names (e.g., "Microsoft" and "Microsoft, Inc.") are not matched by our current techniques; methods that perform such matching [3] have been shown to be useful in information retrieval. In any case, the failure of the additional linguistic features to improve per-

Feature combination	Intercept	Words	LinkIT	Nominator
Words + LinkIt	-3.2850	25.9389	-1.1403	N/A
Words + NominatorAll	-3.2402	22.4196	N/A	5.8726
Words + NominatorPPOU	-3.2440	22.4787	N/A	4.6294
Words + NominatorPPO	-3.2479	22.4412	N/A	4.1257
Words + LinkIt + NominatorAll	-3.2476	23.5958	-1.3459	5.8944
Words + LinkIt + NominatorPPOU	-3.2519	23.7355	-1.4445	4.6586
Words + LinkIt + NominatorPPO	-3.2560	23.7206	-1.4692	4.1479

Table 1: Combinations of features and weights estimated for the corresponding log-linear model.



(a) **(b)**

Figure 3: Topic- **(a)** and story-weighted **(b)** `cdet` as a function of the threshold τ (`tf.idf` weights; groupwise-average clustering; optimal log-linear combination of *Words* with other features).

formance when used alone does not mean that they cannot contribute to better clusterings when used together with the original word vectors, as we show in the next paragraph.

Choice of document feature combinations:

To combine the similarity values obtained by the above methods, we applied the log-linear model of Section 4. We selected 1,358 documents from the training part of the TDT2 corpus for which their topic assignment was known (i.e., manually annotated by the Linguistic Data Consortium). Among the $\binom{1358}{2} = 921{,}403$ pairs of these documents, we randomly selected 100,000 pairs for training models that included different combinations of input features. These combinations and the weights assigned to each similarity vector are shown in Table 1.

We note several interesting observations from Table 1. First, and in line with the empirical results of the preceding paragraph, the weights for words are invariably larger than either of the other two features, and the weights for Nominator are larger (in absolute values) than those assigned to LinkIt. This indicates that, with our current extraction techniques, words remain the most important feature[6], and that the information provided by Nominator is more useful than the simple noun phrase head matching performed by LinkIt. Second, the LinkIt vector is assigned negative weights; this does not mean that noun heads are not useful as a matching feature, but rather indicates that given the information from words and proper nouns, additional matches of noun phrases are evidence for document *dissimilarity*. This surprising result can form the basis of a more detailed analysis of matching noun phrases in the future. Finally, the negative intercept confirms that, in the absence of other information and given the low expected similarity between any two documents[7], it is far more likely for two documents to belong in different clusters than the same cluster. Note that the automatic modeling procedure has explored a range of weights usually not covered by other techniques, which would have been unable to detect the negative correlation of the LinkIt-based similarities with the overall document similarity.

Figure 3 contrasts the performance of the best single feature (*Words*) with models involving additional linguistically motivated features; as Figure 3 shows, the combined features slightly outperform the *Words* feature alone, and extend the range of threshold values for which the `cdet` curve remains low (which is important when the method is applied to unseen data, where a different optimal τ may apply).

5.3 Comparisons with TDT2 results

Having explored different clustering algorithms and assessed the contributions of features and feature combinations on (part of) the TDT2 training set, we selected the best feature combination

[6] This was confirmed with a separate analysis of variance study.

[7] The average similarity was 0.039 for *Words*, 0.030 for *LinkIt*, and 0.012 for *NominatorAll*; accordingly, our model combining these three features predicts that two documents will be in the same cluster 9.14% of the time.

		Optimized for Story-Weighted		Optimized for Topic-Weighted	
		Story-Weighted `cdet`	Topic-Weighted `cdet`	Story-Weighted `cdet`	Topic-Weighted `cdet`
Groupwise-Average Clustering	Training	0.0068	0.0034	0.0083	0.0032
	Test	0.0043	0.0042	0.0046	0.0034
Single-Pass Clustering	Training	0.0078	0.0043	0.0087	0.0034
	Test	0.0072	0.0046	0.0079	0.0036

Table 2: Training and test detection costs for groupwise-average and single-pass clustering (`tf.idf` weights; optimal log-linear combination of *Words*, *LinkIt*, and *NominatorAll*).

		Optimized for Story-Weighted		Optimized for Topic-Weighted	
		Story-Weighted `cdet`	Topic-Weighted `cdet`	Story-Weighted `cdet`	Topic-Weighted `cdet`
Groupwise-Average Clustering	Training	0.0071	0.0033	0.0071	0.0033
	Test	0.0052	0.0037	0.0052	0.0037
Single-Pass Clustering	Training	0.0077	0.0047	0.0090	0.0032
	Test	0.0074	0.0049	0.0084	0.0039

Table 3: Training and test detection costs for groupwise-average and single-pass clustering (`tf.idf` weights; *Words* as the single feature used).

Organization	Story-Weighted `cdet`	Topic-Weighted `cdet`	Average
BBN	0.0040	0.0047	0.00435
IBM	0.0046	0.0042	0.00440
Dragon	0.0045	0.0048	0.00465
UMass	0.0040	0.0064	0.00520
UPenn	0.0071	0.0063	0.00670
CMU	0.0077	0.0057	0.00670
CIDR	0.0096	0.0084	0.00900
UIowa	0.0130	0.0095	0.01125
Our system (a)	**0.0072**	**0.0046**	**0.00590**
Our system (b)	**0.0079**	**0.0036**	**0.00575**

Table 4: The `cdet` values over the test corpus for the TDT2 participants and for our system. The last two rows show the test results that we obtained by using single-pass clustering and the best parameters learned during training for **(a)** lowest story-weighted `cdet`, and **(b)** lowest topic-weighted `cdet`.

(*Words* plus *LinkIt* plus *NominatorAll*) and fixed the threshold parameter τ, both on the basis of the training set. Then we applied both the groupwise-average and single-pass methods on the TDT2 test set. The results, shown in Table 2, reveal that our system not only is stable when applied to a different set of unseen documents but also generally *improves* the cost of detection on the test set (i.e., has a lower cost for the test set than for the training set). Note that Table 2 has separate entries for the threshold values that minimized on the training set the story-weighted (micro-averaged) `cdet` and the topic-weighted (macro-averaged) `cdet`.

Table 3 shows how the original feature, *Words*, fares compared to the combination of all three features for which `cdet` scores were given in Table 2. We observe that the combination of the three features continues to slightly outperform words alone in the test set, as was the case for the training set. Overall, in 11 of the 16 cases presented in Tables 2 and 3 the combined features approach performs better than just using words.

Finally, we compare the performance of our system (with all three features and using the single-pass technique to ensure a fair comparison) on the test set with the performance of the TDT2 participants on the same set of documents (Table 4). Our system places in the middle range of the participants in terms of micro-averaged detection cost (whether optimized for micro- or macro-averages), but performs especially well in terms of macro-averaged detection cost: it places second when optimized for micro-averaging, and first when it is also optimized for macro-averaging.

6 Conclusions

Our analysis has established that both the chosen clustering algorithm and the input features make a difference in the performance of a system that separates documents in topical groups. We confirmed that groupwise-average performs best in the TDT2 setting, and discovered that single-pass can offer a cheaper but close second if the clustering threshold is carefully selected. In addition, we found that using information about matching noun phrase heads and proper nouns can help improve overall clustering quality. We addressed the hard problem of combining similarity values from multiple input features (or different similarity functions) by proposing a theoretically justified statistical model, which was shown in practice to perform as well as or better than exhaustive search.

Of course, the improvement offered by the linguistic features was relatively small, a fact that we believe is due in part to errors made by the extraction tools and in part to the generality of the classes considered. Our motivation for including the noun phrase head and proper name vectors in the first place was to focus the input features on the protagonists in the documents. Taking all noun phrase heads or all person and organization names is only an approximation, and in future work we will explore additional techniques so that only a few proper names or nouns are selected as the major participants for each event described in a document. Similarly, we plan to refine our technique for identifying the location of an event, and incorporate time information, which has already been used with some success by other systems [13, 22].

Acknowledgments

We would like to thank Stefan Negrila and Kazi Zaman, who participated in earlier work that formed the precursor of the experiments reported in this paper. Kathy McKeown, Shi-Fu Chang, and the other members of the STIM-1 research group at Columbia University provided valuable feedback during the design of the reported experiments. The work reported here was funded in part by a National Science Foundation STIMULATE grant, IRI-96-19124. Any opinions, findings, or recommendations are those of the authors, and do not necessarily reflect the views of the NSF.

References

[1] J. Aberdeen, J. Burger, D. Day, L. Hirschman, P. Robinson, and M. Vilain. MITRE: Description of the Alembic system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, 1995.

[2] D. M. Bates and D. G. Watts. *Nonlinear Regression Analysis and its Applications*. Wiley, New York, 1988.

[3] W. W. Cohen. Integration of heterogeneous databases without common domains using queries based on textual similarity. In *Proceedings of the 1998 ACM International Conference on Management of Data (SIGMOD'98)*, June 1998.

[4] J. Fiscus, G. Doddington, J. Garofolo, and A. Martin. NIST's 1998 Topic Detection and Tracking evaluation (TDT2). In *Proceedings of the 1999 DARPA Broadcast News Workshop*, pages 19–24, Herndon, Virginia, February–March 1999.

[5] W. B. Frakes and R. Baeza-Yates, editors. *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[6] L. S. Gay and W. B. Croft. Interpreting nominal compounds for information retrieval. *Information Processing and Management*, **26**(1):21–38, 1990.

[7] Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of the 19th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR-96)*, August 1996.

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

[9] Mark Liberman. Topic Detection and Tracking Principal Investigators meeting, 1998.

[10] Stephen A. Lowe. The beta-binomial mixture model and its application to TDT tracking and detection. In *Proceedings of the 1999 DARPA Broadcast News Workshop*, pages 127–131, Herndon, Virginia, February–March 1999.

[11] K. McKeown, J. Klavans, V. Hatzivassiloglou, R. Barzilay, and E. Eskin. Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-99)*, pages 453–460, Orlando, Florida, July 1999.

[12] National Institute of Standards and Technology. The Topic Detection and Tracking Phase 2 (TDT2) evaluation plan, 1998. Version 3.7, August 3rd, 1998. Available from `http://www.itl.nist.gov/iaui/894.01/tdt98/doc/tdt2.eval.plan.98.v3.7.pdf`.

[13] Ron Papka, James Allan, and Victor Lavrenko. UMass approaches to detection and tracking at TDT2. In *Proceedings of the 1999 DARPA Broadcast News Workshop*, pages 111–116, Herndon, Virginia, February–March 1999.

[14] C. Sable and V. Hatzivassiloglou. Text-based approaches for the categorization of images. In *Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries (ECDL-99)*, Paris, France, 1999.

[15] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, **24**(5):513–523, 1988.

[16] G. Salton and M. Smith. On the application of syntactic methodologies in automatic text analysis. In *Proceedings of the Twelfth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-89)*, 1989.

[17] T. J. Santner and D. E. Duffy. *The Statistical Analysis of Discrete Data*. Springer-Verlag, New York, 1989.

[18] Alan F. Smeaton. Progress in the application of natural language processing to information retrieval tasks. *The Computer Journal*, **35**(3):268–278, 1992.

[19] N. Wacholder. Simplex NPs clustered by head: A method for identifying significant topics in a document. In *Proceedings of the COLING/ACL Workshop on the Computational Treatment of Nominals*, pages 70–79, Montreal, Canada, October 1998.

[20] N. Wacholder, Y. Ravin, and M. Choi. Disambiguation of proper names in text. In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing (ANLP-97)*, pages 202–208, Washington, D.C., April 1997.

[21] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd ACM International Conference on Research and Development in Information Retrieval (SIGIR-99)*, pages 42–49, Berkeley, California, 1999.

[22] Y. Yang, T. Pierce, and J. Carbonell. A study on retrospective and on-line event detection. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-98)*, pages 28–36, Melbourne, Australia, August 1998.