Topic-Based Clustering of News Articles

Najaf Ali Shah
University of Alabama at Birmingham

Ehab M. ElBahesh
University of Alabama at Birmingham

ABSTRACT
Recent years have witnessed an explosion in the availability of news articles on the World Wide Web. Although search-engines' algorithms have made it easier to locate these documents, they still require considerable effort on the part of the user since most search engine algorithms look for keywords and do not take the contents of the entire article into context. We propose a system that clusters articles based on their topics. More specifically, we have focused on applying text mining methods to help solve the problems faced by a media organization or public relations department.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *clustering, Information filtering, Selection process.*

General Terms
Data mining, Text Mining, Clustering.

Keywords
Keywords: article clustering, stemming, Jaccard coefficient

1. INTRODUCTION
The issue of identifying documents that might be of potential interest is increasingly being addressed by Data Mining methods. Specifically, a specialized form of Data Mining, called Text Mining, has been used for tasks such as identifying trends in text documents. TopCat (TopicCategories), for example, is a technique for identifying topics that recur in articles. Representing an article as a set of entities, it identifies related groups of items [1]. By contrast, the system we propose is much less extensive and more practical in that it aims only to solve the problem of grouping together articles of similar topic. News organizations would like to be able to access related documents with minimum effort. Current searching algorithms put too much emphasis on document popularity and other factors uncharacteristic of a particular document.

The system we propose makes this task drastically easier by offering information to the user. Rather than basing searches on specific keywords or popularity, our approach clusters similar articles together based on the content of the entire article. Our text mining program automates the grouping of related news articles. This paper describes the article clustering system we developed. It is meant to run on a database of articles written by a news organization. For this system to produce valuable results, it needs a very large number of articles unless the articles are pre-confined to a narrow domain of topics. This system maintains clusters of documents that are similar to each other.

ACMSE 04, April 2–3, 2004, Huntsville, Alabama, USA.

2. CLUSTERING NEWS ARTICLES

2.1 Data Collection and Cleansing
The data we used consisted of news articles retrieved from the *UAB Reporter*'s archives. The data was collected in records, which were then manipulated by the program.

Data cleansing was necessary for producing vital results. The first step was to remove the stopwords from each article. A stopword occurs so often that it creates no significance to a particular document For instance, the removal of the article "a" and the word "however" does not hinder uniqueness regardless of how often they appear. The list of stopwords used by our program was built by upon a list obtained from [3]. The second step in data cleansing was converting the resulting documents, containing nonstop words, into stemmed words. We used the Porter stemming algorithm which is a process for removing the commoner morphological and inflexional endings from words in English [2]. The reason for this is, for example, to prevent the program from treating the word "brain" and "brains" differently. Figures 1 and 2 illustrate the stopword deletion and stemming that occurs in the program.

> Researchers have discovered a trio of genes that help provide the complex origins of psoriasis, the itchy skin disease triggered when the immune system runs amok.

Figure 1. A typical article used by the program

> research discov trio gene like help provid complex origin psoriasi itchi skin diseas trigger immun system run amok

Figure 2. Stopwords deleted and article stemmed

The final step involved the gathering of the *n* most frequently occurring words in each article. This constraint better characterizes the article. In practice, the words that depict the topic tend to occur more frequently. By the time this part of the program terminates it has produced a listing of the *n* most frequently occurring words in each article.

2.2 Similarity Measure
Our method computes similarity by dividing the number of words that are present in both set A and set B by *n*, the number of most frequently occurring words used. Thus, two articles that have no frequently occurring word in common will have a similarity measure of zero, whereas two articles that have the exactly the same frequently occurring words will have a similarity measure of one. The distance between two articles is the value of the similarity measure subtracted from 1.

2.3 K-Nearest Neighbor Clustering Algorithm

For each article, this clustering method finds the *k* articles that are closest to it, i.e. the articles that are least in distance compared to other articles.

The algorithm starts out by selecting an article, *a*. It then scans the distance matrix for the article, *x*, that has the minimum distance from *a*. An edge is drawn between *x* and *a*. The algorithm proceeds in a similar fashion, finally looking for the *k*th nearest neighbor and adjusting the graph accordingly.

The procedure described in the previous paragraph is repeated for each article. By the time the scanning part of the algorithm terminates, it has produced an undirected graph with a vertex corresponding to each article. Each connected component of the graph is then treated as a cluster. For example, if there is an edge between *x* and *y* and an edge between *y* and *z*, all three vertices are said to be part of the same cluster even if there is no edge present between *x* and *z*.

2.4 Single-Link Clustering Algorithm

This clustering approach requires the user to input a maximum distance value, ε. The algorithm scans the distance matrix and if it finds a distance that is less than or equal to ε, it draws an edge between the two vertices corresponding to the two articles. By the time the scanning concludes, the algorithm has produced an undirected graph with a vertex corresponding to each article.

For example, if this algorithm is run with an ε value of 0.7 and an *n* value of 10, it will construct an edge between any two articles that have at least three words in common between their *n*-most frequently occurring words list. The algorithm then proceeds to print the clusters from the graph it produced. As in the K-Nearest Neighbor Algorithm, each connected component of the graph is treated as a cluster.

2.5 Hybrid Algorithm

This algorithm is very similar to the k-nearest neighbor algorithm with only one major difference. This algorithm draws an edge between two nearest neighbors only if the distance between these two neighbors is less than or equal to ε, the value of which is entered by the user.

3. RESULTS

We tested the Single-Link algorithm on fifteen hundred articles, with an ε value of 0.7. It accomplished its goal reasonably well by getting meaningful clusters. For instance, in our testing, this algorithm successfully clustered stories about the UAB football team that were centered on a specific player. Although Single-Link did not output meaningless clusters with a stringent value of ε (for e.g., 0.5), it certainly has the potential of doing so. For instance, if we run it with a high value of ε such as 0.9, this algorithm can form a chain and cluster articles which have seemingly nothing in common. In our testing on 1500 articles and ε=0.7, the largest cluster this algorithm produced had over 200 articles within it, most of which had nothing in common. Conversely, if one has a very large amount of data, one can decrease the ε value to 0.5 or less to get tighter, more meaningful clusters.

ACMSE 04, April 2–3, 2004, Huntsville, Alabama USA

K-Nearest Neighbor didn't perform well since it grouped together articles that had very little or nothing in common. This is because for a particular article, this algorithm simply looks for the article closest to it and puts them in the same cluster even if the distance between these two articles is almost one (denoting no similarity). In other words, this algorithm *always* groups a particular article with another article. In our experiments, this algorithm produced numerous similar-sized clusters, the articles within which had little to nothing in common. Thus, we gather that this algorithm is not well-suited to this purpose.

The Hybrid algorithm clearly outperformed the other algorithms. Even with ε values as high as 0.7, we were able to get meaningful clusters. The quality of this algorithm proved to be far better than that of the other algorithms because it did not group together articles that had nothing in common. For example, this algorithm grouped together articles about a research study on cholesterol that were written over a span of several months. As was clearly visible from our results, this algorithm is a much safer choice than Single-Link because it is less likely to form a chain given large amounts of data. It is important to note here that the ε of this algorithm and the ε of Single-Link do *not* perform the same function. In Single-Link, if the distance between two articles is less than or equal to ε, the two articles are clustered together. Thus in this case ε judges the similarity between two articles. In the K-Nearest Neighbor with Constraint algorithm, however, the ε is used simply to keep the algorithm from clustering articles that are not related to each other. In other words, it protects against adding outliers.

Table 1. Summarized Results of Clustering 1500 articles (*n*=10, *k*=1)

Algorithm	ε	No. of non-singleton clusters	Avg. No. of elements in non-singletons
Single-link	0.7	31	42.16
Single-link	0.5	132	2.5
Nearest Nbr.	n/a	277	5.42
Hybrid	0.7	277	4.72
Hybrid	0.5	139	2.37

4. REFERENCES

[1] C. Clifton and R. Cooley. Topcat: Data mining for topic identification in a text corpus. 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases, Prague, Czech Republic, Sept. 15–18 1999.

[2] M. Porter, "An algorithm for suffix stripping," Automated Library and Information Systems, vol. 14, no. 3, pp. 130-137, 1980.

[3] *ERIC Database: Searching Assistant--Stopwords.* http://www.askeric.org/Eric/Help/stop.shtml

5. ACKNOWLEDGEMENTS

We would like to thank Dr. Alan Sprague of the UAB Computer Science department for his invaluable support.