

# **Protégé: Biomedical Topic Detection and Tracking**

Jerry Ye  
School of Information  
UC Berkeley

Email: [jerrye@ischool.berkeley.edu](mailto:jerrye@ischool.berkeley.edu)

Jih-Yin Chen  
School of Information  
UC Berkeley

Email: [jimmy@ischool.berkeley.edu](mailto:jimmy@ischool.berkeley.edu)

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Motivation . . . . .	2
1.2	Topic Detection and Tracking . . . . .	3
1.2.1	Topic Detection . . . . .	3
1.2.2	Topic Tracking . . . . .	3
<b>2</b>	<b>Algorithm</b>	<b>4</b>
2.1	K-means . . . . .	4
2.2	Feature Selection . . . . .	5
2.3	PCA . . . . .	5
<b>3</b>	<b>Implementation</b>	<b>7</b>
3.1	System Architecture . . . . .	7
3.2	WEKA . . . . .	8
<b>4</b>	<b>Conclusions</b>	<b>9</b>
4.1	Results . . . . .	9
4.2	Conclusion . . . . .	10
4.3	Future Work . . . . .	10

# Chapter 1

## Introduction

NOTE: Portions of this project and write-up were done in conjunction for IS256: Applied Natural Language Processing and submitted for credit.

### 1.1 Motivation

Topic detection and tracking (TDT) is the study of detecting topics and tracking the evolution of those topics in a constantly updated collection.[1] The definition of a topic is broad in that it can be a subject of interest or event that takes place. The purpose of the project is to enable users to search for documents relevant to the topics they're interested in researching. Upon browsing these, they are provided with the option of tracking the particular cluster(s) that their selection falls in. This project's relevance to natural language processing extends to the areas of feature selection and text classification. For our particular case, we chose to use biology research papers as our collection for analysis but TDT can be applied to several other domains such as news, academia, and domestic spying. When users register to use our product, they are prompted with a search box that allows them to find relevant documents. The documents are clustered by color which provides for easy recognition of the different groupings. Every article is shown with its title, description, URL, and date of publication. When the users select the document(s), they are presented with all the documents that are in the same clusters as those they selected. Also included with every article is the ability to click Track it which records what interests the users so that in the future, they have the option of following a particular research topic as it develops. Articles that are recently published are conveniently tagged with a new image to tag articles that have recently come in.

For our document collection, we took RSS feeds from BioMed Central(<http://www.biomedcentral.com>). The website is an independent publishing house aggregating peer-reviewed biomedical research papers. We chose this domain as a starting corpus because of the availability of the text but any genre would have sufficed.

## 1.2 Topic Detection and Tracking

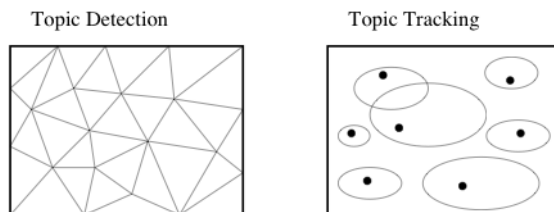


Figure 1.1: Clusters for topic detection and tracking.

### 1.2.1 Topic Detection

Topic detection is the process of grouping documents with similar topics into the same cluster. In Figure 1.1, we can see that topic detection assigns "hard" clusters to each document, meaning that a document can only belong to one cluster. Detection takes place in two phases, once with initial training data and again with online updates. In the initial topic detection phase, we perform feature selection on the training data and end up with a feature vector for each document in the collection. We then perform K-means clustering on the document set and end up with  $k$  clusters, each representing a unique topic. The centroids, the means, of each cluster are then used to compare with future documents.

In online topic detection, there are a number of processes that must take place to identify new topics or topic membership.[2] We first take a research paper and compute the feature vector using the same feature set selected during training. Given the feature vectors for each document, we compare document vectors to the centroids of existing clusters and compute a similarity score. For detection, we classify a document as belonging to an existing cluster if the similarity score is high and as a new topic if the score falls below some threshold.

### 1.2.2 Topic Tracking

Topic tracking is an online process where documents from the RSS feed are downloaded and processed into feature vectors for classification. In traditional topic tracking, each topic cluster is used to train a classifier to detect topic membership. As demonstrated by Franz[4], our approach streamlines this by utilizing the centroids of each cluster to do the classification. For tracking, we compute the distance between the centroids and pick the one with the closest similarity. We then label the document as being the same topic and update the centroids. In Figure 1.1, we see that topic detection is a soft classification task that can yield multiple labels for one document. Although our current implementation only picks the top cluster above a threshold, we can easily extend it to choose the top  $n$  clusters. Another method for tracking would be to do Latent Dirichlet Allocation for soft clustering.[3]

# Chapter 2

## Algorithm

### 2.1 K-means

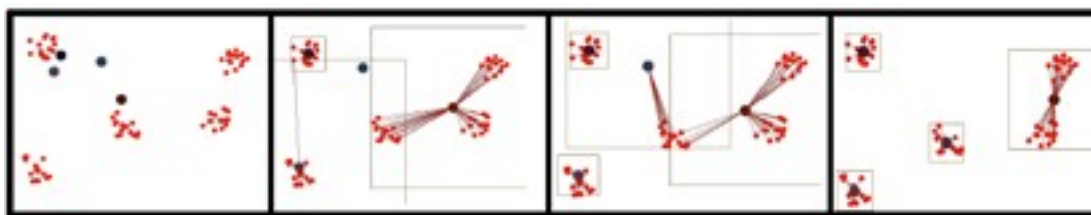


Figure 2.1: K-means clustering with 4 initial clusters.

Topic detection and tracking is achieved using K-Means algorithm. K-Means is an unsupervised clustering algorithm that clusters objects based on attributes into  $k$  partitions. The objective is to minimize total intra-cluster variance, or, the squared error function:

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} |x_j - \mu_i|^2 \quad (2.1)$$

where there are  $k$  clusters  $S_i, i = 1, 2, \dots, k$  and  $\mu_i$  is the centroid or mean point of all the points. The algorithm starts by partitioning the input points into  $k$  initial sets, either at random or using some heuristic data. It then calculates the centroid of each set. It constructs a new partition by associating each point with the closest centroid. Then the centroids are recalculated for the new clusters, and algorithm repeated by alternate application of these two steps until convergence, which is obtained when the points no longer switch clusters (or alternatively centroids are no longer changed). Figure 2.1 illustrates the process of K-means clustering. Figure 2.1(a) shows the initial randomized centers and a number of points. In 2.1(b), centers have been associated with the points and have been moved to the respective centroids. In 2.1(c), the association is shown in more detail, once the centroids have been moved. In 2.1(d), the centers are moved

to the centroids of the corresponding associated points.

The K-Means algorithm is applied to topic detection and tracking. Each cluster is seen as a single topic, and the data points it consists of represent the papers for that particular topic. When a new paper comes into the system, it compares the distances to each centroid and finds the closest one. A new topic is detected if the closest distance is above a given threshold, which means the paper is not similar to each topic in the corpus. Consequently, a new cluster (topic) will be created starting with this paper. Similarly, an existing topic is tracked if the closest distance is below the threshold, which means the paper is similar enough to that cluster. The paper will join that cluster (topic).

## 2.2 Feature Selection

For the biology research papers, we decided to use the words in the title and abstract as our primary sources of features. The articles found at our web source had these two sections readily available for extraction. With them, we tokenized the words into features for analysis. In addition to the unigrams attained, we also took into account bigrams. Our justification for this was that bigrams are often more meaningful than the unigrams alone. As an example, Vitamin A as a bigram would prove more useful to the user than either Vitamin or the capital letter A separately. Weights for these were assigned accordingly: unigrams with weights one, and bigrams with weights two. Those features found in the title were given more weight, specifically +5, as we noted that the title is concise and does a good job summarizing the overall paper. A generic stoplist was enhanced by adding medical terms that were not useful for feature selection.

Stemming was implemented as we felt that words with different stems should not be treated differently if they are in different documents; we used a Porters stemmer on our collection. The next step involved the selection of the top 1000 features for processing. We decided this was a good number as attempts to use the top 5000 features posed quite a challenge to analyze and process on our computers. PCA took over 12 hours and did not finish its processing with the 5000 features but having only 1000 features allowed us to view results within 4 hours. Finally, we decided to implement tf.idf weighting on words to prevent those features that occur in the majority of documents from having too much weight. We aim to put weight on those terms that are found clustered in a few documents as opposed to the whole collection.

## 2.3 PCA

In feature selection, we focused on two aspects of feature use: 1. Select more features. 2. Pick out the good features. The first aspect is intuitive whereby the more features we have, the better we can describe the text, and thus we will get better performance on clustering. For the second aspect, we try to eliminate some noisy or outlier features and retain the good features that are truly helpful for clustering. This is more difficult to achieve than simply using more features. We wanted to investigate how to select good features

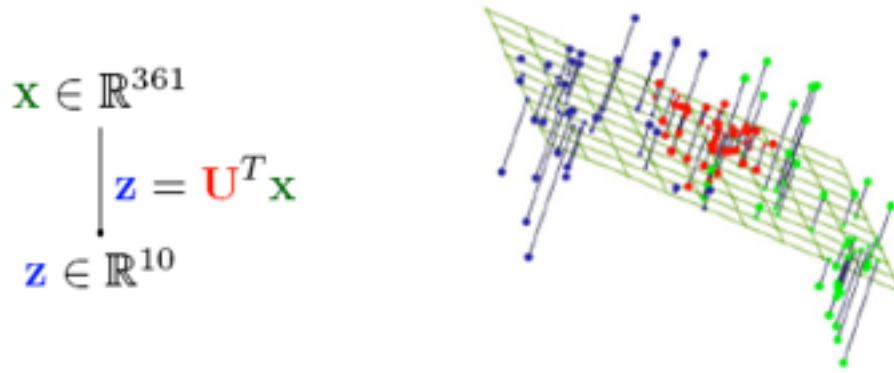


Figure 2.2: Principle Component Analysis projects data into a lower dimensional subspace.

to improve clustering performance.

We used dimensionality reduction to reduce the dimensionality of the features. This is achieved by principal components analysis (PCA). PCA is a technique for simplifying a dataset, by reducing multidimensional datasets to lower dimensions for analysis. It can be used for dimensionality reduction in a dataset while retaining those characteristics of the dataset that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. Such low-order components often contain the "most important" aspects of the data. PCA selects eigenvectors with highest eigenvalues. The number of significant eigenvalues will be the reduced dimensionality.

Figure 2.2 shows the dimensionality reduction by PCA. The original dataset  $X$  is reduced to a lower dimensionality by multiplying by the transpose of its principal component  $U$ . The right part of Figure 2.2 shows the projection of original data to a subspace using PCA. In our project, we have initial set of 1000 text features. We run PCA to reduce the feature dimensionality to about 820. Each projected new feature is a linear combination of the original features. The influence of the reduced feature dimensionality on the clustering performance will be shown in the later part of this paper.

# Chapter 3

## Implementation

A demonstration of our system can be accessed at the following address: <http://cocobean.gotdns.com/protej>

### 3.1 System Architecture

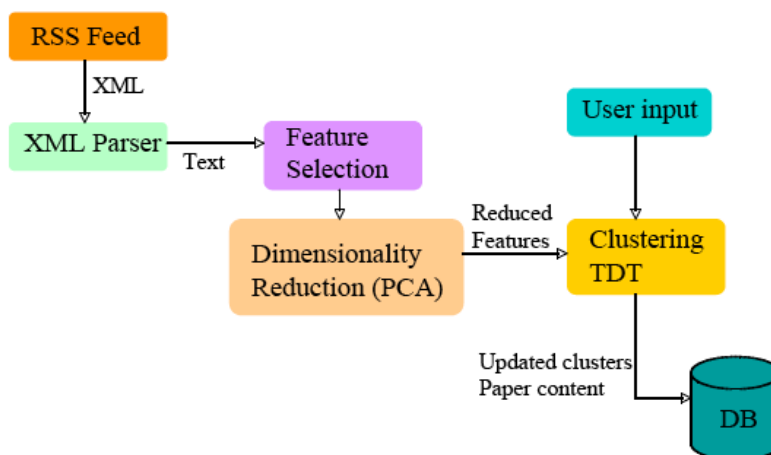


Figure 3.1: System architecture.

The system retrieves data from BioMedCentral RSS feeds every 30 minutes to update the biomedical papers. There is an XML parser that extracts text from the XML documents. Upon extracting the title and abstracts of the paper, the text undergoes tokenization and each token is regarded as a feature of the paper. The tokens are sent to the feature selection component, where words are processed with stemming, frequency trimming, and stop word removal. We picked the top 1000 features of the corpus after ordering them by tf.idf. The dimensionality reduction component reduced the feature dimensionality using PCA. PCA reduced the feature size to about 820 features. We tried both the original and reduced size of features to experiment on the effects of dimensionality reduction. The paper, represented by a vector of features,

then goes to the clustering component. There are two possible consequences in the clustering process. For topic detection, the system generates a new cluster and its centroid; for topic tracking, the paper joins an existing cluster and the centroid of that cluster is updated accordingly.

The information about the paper along with the updated/new centroids are then stored into a database. For the user interface, users can specify which topic they want to track. Each user's tracking information is sent to the clustering component. When a new paper is clustered into the topic specified by the user, the system will send the new paper to the user via email.

### 3.2 WEKA

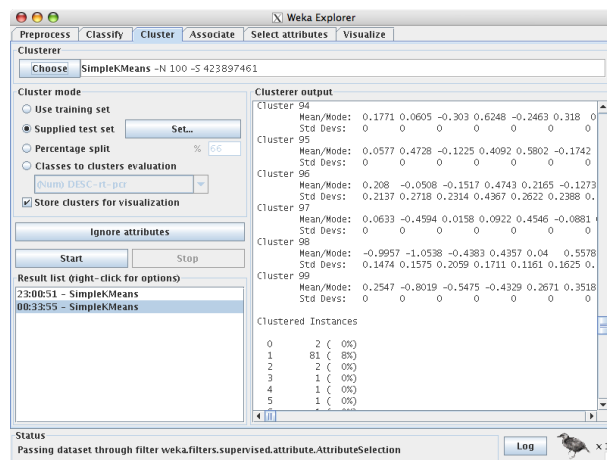


Figure 3.2: Clustering using K-means in WEKA.

Weka was used in the project to perform feature selection and clustering. In the system architecture diagram shown in Figure 3.1, Weka was used to perform dimensionality reduction using PCA as well as the initial clustering using K-means. Although we attempted to run other clustering algorithms like EM on the data, we abandoned our attempts due to extremely long running times with little improvements in cluster quality. Work in Weka consisted of clustering the documents with K-means using parameter searching through the number of clusters and random seeds to limit the number of documents in a single cluster below a certain percentage.

Once the initial clusters were computed, we outputted the centroids and the clustered data into MySQL databases for the frontend. Instead of running clustering again in Weka for topic tracking, we used the centroids to compute similarity scores and updated them in the database as more documents are clustered.

# Chapter 4

## Conclusions

### 4.1 Results

After clustering on the initial data, we explored some of the clustered results and found that most had similar terms and some clusters even shared common topics. In the selected passage below, we see that the papers are related to types of cancers afflicting females.

Bcl-2 protein expression is associated with p27 and p53 protein expressions and MIB-1 counts in breast cancer

Background: Recent experimental studies have shown that Bcl-2, which has been established as a key player in the control of apoptosis, plays a role in regulating the cell cycle and proliferation. The aim of this study was to investigate the relationship between Bcl-2 and p27 protein...

<http://www.biomedcentral.com/1471-2407/6/187>

Mismatch repair and treatment resistance in ovarian cancer

Background: The treatment of ovarian cancer is hindered by intrinsic or acquired resistance to platinum-based chemotherapy. The aim of this study is to determine the frequency of mismatch repair (MMR) inactivation in ovarian cancer and its association with resistance to platinum-based chemotherapy. Methods: We determined, microsatellite instability...

<http://www.biomedcentral.com/1471-2407/6/201>

Although not all results were as ideal as in the above cluster, most had at least some common terms.

## 4.2 Conclusion

TDT relies heavily on clustering and consequently shares the inherent problems that comes with clustering. During topic detection with K-means, the initial number of clusters were chosen to minimize the number of papers in a single cluster. However, there was no systematic way of choosing the number of initial clusters besides parameter search. Another problem with TDT is that its reliance on clustering means that documents are grouped together based on mathematical similarity. The problem arises when the quality of features do not enable accurate clustering of data. In this case, mathematical similarity does not necessarily imply topic similarity. Our initial choice of using K-means is mainly due to its relatively good results and efficiency. However, other learners such as EM, k-nearest neighbors[5], or hierarchical clustering[6] algorithms might yield better results at the cost of compute time.

We realized that using Principle Component Analysis for feature reduction was a poor choice since the data is not linear in nature. We approached the problem by attempting to select better features through dimensionality reduction. However, we came to the conclusion that when dealing with sparse textual data, the more features the better.

## 4.3 Future Work

A possible improvement for topic tracking would be to use Latent Dirichlet Allocation (LDA) to classify documents into appropriate clusters. LDA allows for the computation of probabilities of class membership using generative models representing latent topics. Using LDA would allow us to use a more intuitive soft classification tracking method where new documents can be labeled as belonging to multiple topics.

When initializing our clusters for K-means, we had to choose a random seeding of centroids. The initial centroids are then updated to maximize variance of documents in the clusters. A serious problem with K-means arises when multiple centroids are seeded within a true cluster. Updates of the centroids in this case usually results in unintuitive results. A solution to this would be to use K-means++ during our initial clustering phase. K-means++ attempts to seed centroids that are weighted inversely proportional to the distance between other clusters. K-means++ usually results in improved clustering results by eliminating cases where natural clusters have more than one centroid seeded.

# Bibliography

- [1] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report, 1998.
- [2] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *Research and Development in Information Retrieval*, pages 37–45, 1998.
- [3] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation, 2002.
- [4] Martin Franz, J. Scott McCarley, Todd Ward, and Wei-Jing Zhu. Unsupervised and supervised clustering for topic tracking. In *Research and Development in Information Retrieval*, pages 310–317, 2001.
- [5] Yiming Yang, Tom Ault, Thomas Pierce, and Charles W. Lattimer. Improving text categorization methods for event tracking. In *SIGIR*, pages 65–72, 2000.
- [6] Yiming Yang, Jaime Carbonell, Ralf Brown, Tom Pierce, Brian T. Archibald, and Xin Liu. Learning approaches for detecting and tracking news events, 1999.