

**An Approach to Improving Cluster Labeling and Evaluation**

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF MINNESOTA

BY

Anand Mohan Jha

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
MASTER OF SCIENCE

Donald B. Crouch

February, 2014

©Anand Mohan Jha 2014

ALL RIGHT RESERVED

## **Acknowledgement**

Research and thesis in nature is the principal challenge for any student in his graduate study. And completion of it is by far the biggest achievement in my life. I am grateful to Dr. Donald Crouch and Dr. Carolyn Crouch for helping and finalizing my thesis. Dr. Donald Crouch helped and trusted me in tough times. He was always there to motivate and guide me. His motivation and confidence was the greatest force which helped me to be perseverance in completing my thesis.

I am thankful to Dr. Ted Pedersen for providing me his deep insight on the topic. I sincerely thank him for introducing me to this topic and teaching many things in the process. I appreciate Dr. Pete Willemsen for teaching us the intensity of effort we need to sail through the graduate study. I am in debt to Dr. John Greene for teaching me Enumerative Combinatorics and listening to my all the questions and patiently answering them.

My special thanks to Lori Lucia who I felt was there for us like a mother to guide, help, chide and care. She is amazing woman and UMD is great to have such a person. I am also thankful to Clare who always helped and treated us like a friend. I am thankful to Aishwarya, Sadhna, Kiran and others who took enough pain to proofread my writing which I consider the biggest help for anyone in thesis writing.

I would like to thank the UMD Graduate School and the Director of Graduate Studies, Dr. Carolyn Crouch, for the financial support provided to me in support of my research through the Quality Metrics Allocation/Chancellors Graduate Fellowships.

Finally I am thankful to my friends and family who have faith in me that I will complete my thesis.

## Abstract

The clustering of a large document collection produces subsets of documents (typically overlapping) such that documents within a given cluster exhibit substantial similarities with each other. In this work, the final phase of the clustering process is to generate labels for each cluster, that is, a set of terms that represent the inherent meaning associated with a cluster. Although several methods exist for generating labels, little work has been done in developing methods that determine the quality of the labels. In other words, do the labels represent terms that a human might associate with a cluster? Do they enable the user to readily distinguish between clusters? Do they provide insight into the inherent meaning of the documents in the cluster? In this thesis, we focus on developing a tool that automatically assesses the quality of document cluster labels. Our objective is for the tool to be flexible, extensible, and reliable. It uses the Hungarian algorithm [16] to calculate the accuracy of the labels.

We analyze the performance of our evaluation tool using cluster labels generated by the labeling mechanism of SenseClusters [21], a comprehensive package that generates clusters utilizing unsupervised learning. Label generation is based on the selection of the top five or ten bigrams as ranked by a measure of association. Since selecting features is a significant step in generating labels, we extend the labeling mechanism of SenseClusters by incorporating higher valued n-grams and tf-idf term weighting and then analyze the quality of the labels produced by these additional methods. The experimental results indicate that trigram features produce better results than the traditional unigram or bigram features of SenseClusters. Also, using tf-idf improves the quality of terms in the labels over those produced by the similarity mechanism of the SenseClusters.

# Contents

<b>List of Figures</b>	<b>vi</b>
<b>List Of Tables</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>3</b>
2.1 Cluster Labeling: . . . . .	3
2.2 Cluster Labeling in SenseClusters . . . . .	4
2.2.1 Stop Word List . . . . .	5
2.2.2 Frequency Cutoff . . . . .	6
2.2.3 Window Size . . . . .	6
2.2.4 Measure of Association . . . . .	6
2.2.5 Score Cutoff . . . . .	7
2.2.6 Feature Selection . . . . .	7
2.2.7 Cluster and Labeling Examples . . . . .	7
2.2.8 Summary . . . . .	12
2.3 Evaluating Cluster Labels . . . . .	13
<b>3 A Cluster Label Evaluation System</b>	<b>14</b>
3.1 User Options . . . . .	14
3.1.1 SenseClusterLabelFileName . . . . .	14
3.1.2 GoldKeyFileName . . . . .	15

3.1.3	WeightRatio . . . . .	16
3.1.4	IsClean . . . . .	16
3.2	Label Evaluation . . . . .	17
3.2.1	User provides the mapping information . . . . .	18
3.2.2	User does not provide the mapping information . . . . .	19
3.2.3	Calculating the Similarity Score . . . . .	20
3.3	Hungarian Algorithm . . . . .	21
<b>4</b>	<b>Evaluation of the Algorithm</b>	<b>24</b>
4.1	Test Data . . . . .	24
4.2	Experiments . . . . .	24
4.2.1	Case 1: User provides the mapping information and the gold standard keys	24
4.2.2	Case 2: User provides mapping information and Wikipedia is used to generate the gold keys . . . . .	26
4.2.3	Case 3: System establishes mapping information and the user provides the gold keys . . . . .	26
4.2.4	Case 4: System establishes mapping information and Wikipedia is used for the gold keys . . . . .	28
4.3	Summary . . . . .	28
<b>5</b>	<b>N-gram Features Selection</b>	<b>30</b>
5.1	Description . . . . .	30
5.2	Experimental Setup . . . . .	31
5.2.1	Assumption . . . . .	31

5.2.2	Training and testing dataset . . . . .	31
5.3	Results . . . . .	32
5.3.1	Label generation using bigram as the feature set . . . . .	32
5.3.2	Label generation using trigram as the feature set . . . . .	34
5.3.3	Label generation using n-grams as the feature set with $n \geq 4$ . . . . .	34
5.4	Summary . . . . .	36
<b>6</b>	<b>Using Tf-idf for Cluster labeling</b>	<b>37</b>
6.1	Description . . . . .	37
6.2	Experimental Setup . . . . .	38
6.2.1	Algorithm used in Tf-idf Process . . . . .	39
6.3	Code Snippets . . . . .	40
6.4	Results: Bigram and Trigrams with Tf-idf . . . . .	40
6.5	Summary . . . . .	46
<b>7</b>	<b>Conclusion</b>	<b>47</b>
<b>8</b>	<b>Future Work</b>	<b>48</b>
	<b>References</b>	<b>50</b>

## List of Figures

1	A Sample Input Passed to the LabelEvaluation Module . . . . .	17
2	Flowchart for Label Evaluation System . . . . .	18
3	Module to Access Data from Wikipedia . . . . .	19
4	Module to Calculate the Similarity Score . . . . .	22
5	Contingency Matrix and Hungarian Algorithm Result . . . . .	23
6	Code Snippet for Bigram Selection and Frequency Count . . . . .	32
7	Code Snippet for Trigram Selection and Frequency Count . . . . .	33
8	A Typical Document Entry in a Cluster File . . . . .	39
9	Code to Calculate Term Frequency . . . . .	41
10	Code to Calculate Inverse Document Frequent . . . . .	41
11	Code to Calculate TF-IDF . . . . .	42
12	Visualization of ‘Finding the Topics Using Terms of the Labels’ . . . . .	49



## List of Tables

1	Case 1 User Provides the Mapping Information and the Gold Standard Keys . . . .	26
2	Case 2 User Provides Mapping Information and Wikipedia is Used to Generate the Gold Key . . . . .	27
3	Case 3 System Establishes Mapping Information and the User Provides the Gold Keys	28
4	Case 4 System Establishes Mapping Information and Wikipedia is Used for the Gold Keys . . . . .	29
5	Example Showing Unigram, Bigram, Trigram, Quadragram Feature Selection . . .	31
6	Labels Generated Using Bigram Features . . . . .	33
7	Similarity Scores Generated for Each Label Against Each Topic . . . . .	34
8	Contingency Matrix Showing the Diagonal Alignment after Hungarian Algorithm .	34
9	Labels Generated Using the Trigram Features . . . . .	35
10	Similarity Scores Using Trigrams . . . . .	35
11	Hungarian Alignment Using Trigrams . . . . .	36
12	The Calculation of Term Frequency, Inverse Document Frequency and TF-IDF [22]	38
13	Labels Generated by SenseClusters using Bigrams . . . . .	42
14	Labels Generated by TF-IDF using Bigrams . . . . .	43
15	Labels Generated by TF-IDF using Trigrams . . . . .	44
16	Similarity Scores of Labels Assigned using TF-IDF-Trigram, TF-IDF-Bigram and Original SenseClusters Labeling and Final Conclusion of Label Assignment of Various Options . . . . .	45
17	Similarity Scores of Labels Assigned using SenseClusters-Bigram, SenseClusters-Trigram Labeling . . . . .	45



# 1 Introduction

Clustering of document collections is an analytical procedure that allows the categorization and organization of documents into subsets (clusters) such that the documents within a given cluster exhibit substantial similarities to each other. Clusters of objects with similar meaning are generally obtained through a statistical analysis of the dataset. Although clustering provides a means of organizing large data collections, users may not readily be able to determine the inherent meaning of the individual clusters. Cluster labeling [14] [13] facilitates an understanding of the clusters by creating and assigning labels to clusters such that the labels themselves reflect the central meaning of each cluster in the dataset. A good cluster labeling mechanism is critical to a data clustering technique involving the end-user, as this step is responsible for the presentation of data to him/her. It serves as an indicator reflecting the quality as well as the success or failure of the clustering process. In this context, cluster labeling, whether manual or automatic, may be viewed as the final step of the data clustering process.

The traditional approach [12] [20] to identifying the labels for a cluster is to select the prominent and dominating tokens in that cluster, say, for example, the  $n$  most frequently occurring terms or phrases within the cluster. This approach relies completely on the clustered data and provides a statistically important set of words as labels. Although traditional approaches may provide some good terms as labels, they often fail to provide labels that aid in identifying the primary focus of the contents of the cluster. For example, if we were to run a clustering process on a set of newspaper articles containing peripheral information on President Bill Clinton, the kind of labels we expect from the traditional labeling approach might be White House, Georgetown University, Al Gore, Democratic Party, Dont ask, dont tell, of Arkansas. However, redefining the labeling mechanism to produce a phrase as simple as Bill Clinton as a label would vastly improve the readability and understandability of the cluster.

Although several methods exist for generating labels, little work has been done in developing methods that determine the quality of the labels. In other words, do the labels represent terms that a human might associate with the clusters? Do they enable the user to readily distinguish between the clusters? Do they provide the user with the inherent meaning of the documents contained within a

cluster?

In this thesis, we focus on developing a tool for evaluating labels that meet these criteria. We analyze its performance using cluster labels generated by the labeling mechanism of SenseClusters [21], a comprehensive package that generates clusters based on unsupervised learning. Label generation is based on the selection of the top five or ten bigrams as ranked by a measure of association. A bigram is defined as a pair of terms in a sentence bounded by a window of size  $w$ , where  $w$  is an integer whose value is greater than or equal to 2 [21] [12].

The thesis is organized as follows. Chapter 2 contains a brief overview of cluster labeling and the typical procedures for evaluating cluster labeling including SenseClusters. In Chapter 3 we introduce an automatic procedure for evaluating cluster labels and in Chapter 4 run several experiments to test its validity. Since feature selection is a significant step in generating labels, we enhance the label mechanism of SenseClusters by incorporating within it higher valued  $n$ -grams (Chapter 5) and tf-idf term weighting (Chapter 6) and use our algorithm to evaluate the quality of the labels produced by these additional methods. Conclusions and future work are contained in Chapter 7.

## 2 Background

This section provides a brief overview of the previous work on which this thesis is based.

### 2.1 Cluster Labeling:

Cluster labeling is a long-standing problem pursued by psychologists, computer scientists etc. The basic approaches to cluster labeling include: (1) the traditional approach that relies only on the contents of a cluster, and (2) the combined approach that includes additional data about the contents of the clusters, obtained from external informational resources.

The traditional approach uses so-called important terms of the clusters as the labels for the clusters. Important terms might be the most frequently occurring words, the most predictive words, the terms that are nearest to the centroids of the clusters, etc. This approach assumes that the most prominent words of a cluster reflect its semantic content. In practice, this may not always be the case.

The combined approach uses an external application such as Wikipedia or WordNet+ to improve the labels formulated from the contents of a cluster, as follows. The first step is similar to the traditional method, where the  $n$  most prominent terms for each cluster are identified. In the second step, these  $n$  terms are used as arguments to the external application in order to produce better and more readable labels [4].

Consider the example from Chapter 1. Suppose the traditional approach uses the most frequently occurring bigrams to select labels and that the labels generated for the cluster on Bill Clinton are White House, Democratic Party, ..., etc. . Now if we use the combined approach (based on a search of Wikipedia) using these bigrams, we may be able to generate the alternate cluster label, Bill Clinton.

Cluster labeling processes are generally based on either supervised learning or unsupervised learning. Supervised learning [21] [5] is a means of learning about data with the help of training data, which are often manually created and may severely limit the learning process [13]. Unsupervised learning techniques primarily use statistical information from the clustering process to produce labels. Earlier versions of cluster labeling were based on supervised learning such as a dictionary

or knowledge of words created by humans [20]. Later works on cluster labeling were based on unsupervised learning [12] [13].

Popescul and Ungar [20] developed two different methodologies for labeling documents. In one method they assumed the existence of a document hierarchy and used the  $\chi^2$  test of significance to determine the labels that are spread across the hierarchy. In their second method, they used the set of terms that were both most frequently occurring and most predictive to uniquely identify a cluster. Their results produced better labels than those generated using only the most frequently occurring or the most predictive terms.

Carmel, Roitman and Zwerdling [4] from IBM Research Israel proposed using Wikipedia to improve the cluster labeling mechanism. They used two independent approaches to find the labels. One is based on unsupervised learning. The other uses Lucene indexing to extract important terms from the text of the clustered documents. The important terms are then used to get candidate labels from Wikipedia. Finally both sets of labels are judged, and the better of the two is presented as the labels for the cluster. They claim their results match the human-assigned labels in 85% of the cases.

Another unsupervised labeling process is described by Kulkarni and Pedersen [13]. It uses SenseClusters, originally developed by Purandare and Pedersen [21]. SenseClusters is based on unsupervised learning. It extracts the lexical features from the data to be clustered and uses these features to represent context. Contexts are created using either first or second order context representation. These contexts are clustered using a standard clustering algorithm. Kulkarni and Pedersen use the top five or ten bigrams of a cluster as the descriptive label of the cluster. The terms from this descriptive label which are unique to the cluster are described as the discriminating label.

Since we plan to evaluate labels generated by SenseClusters using the algorithm developed in this thesis, we include a detailed description of SenseClusters and some examples of its use.

## **2.2 Cluster Labeling in SenseClusters**

The cluster labeling mechanism of SenseClusters [14] uses a traditional approach for creating labels. As previously noted, this mechanism produces two kinds of labels for each cluster, a descriptive label and a discriminating label [21]. The descriptive label contains the top bigrams of the cluster

as identified by a statistical method that measures the degree of association between words (for example, log-likelihood). The discriminating label consists of terms in the descriptive label that do not appear in the descriptive label of any other cluster. The discriminating label is intended to capture information that uniquely represents the cluster and to yield more precise information about the cluster than that provided by the descriptive label. For example, if the descriptive labels for two clusters are as follows:

*Cluster 0: "George Bush, Al Gore, White House, George W, Britain London, U S, Prime Minister, New York"*

*Cluster 1: "U S, Al Gore, White House, George W, York Times, New York, Prime Minister, President B\_T"*

the discriminating labels for these clusters are "*George Bush, Britain London*" for Cluster 0 and "*York Times, President B\_T*" for Cluster 1. Note that the term *B\_T* serves as a placeholder in the data for the proper names Bill Clinton or Tony Blair.

The SenseClusters labeling mechanism is a function of several parameters, such as stop words, measure of association, window size, frequency cutoff for feature selection, etc. We briefly describe these parameters and then illustrate their impact using three different test cases.

### **2.2.1 Stop Word List**

A stop word list is the collection of words that are considered non-meaningful or non-substantive with regard to the text, such as *the, this, that, is, was, were*, etc. In SenseClusters, a user can provide his/her own file of stop words, use an existing stop word list, or not use a stop list. In the later case, stop words may appear in the cluster labels. Although such words are not very useful in themselves and may obstruct the actual meaning of a label, removing stop words can prevent the identification of phrases, a very crucial part of any language.

### 2.2.2 Frequency Cutoff

To ignore pairs of words which co-occur by chance, the user specifies a threshold value. All the bigrams having a frequency lower than this value are ignored. The frequency cutoff default value is 5.

### 2.2.3 Window Size

The window size restricts the selection of bigrams. When selecting labels, words that lie outside the window boundary will not be part of the bigram set. For example, consider a sentence containing the phrase *New York Times headline* with a window size of 3. The bigrams generated are {*New York, New Times, York Times, York headline, Times headline*} . The bigram {*New headline*} will not be included. The default window size defined by SenseClusters is 2. In this case, consecutive words are considered bigrams.

### 2.2.4 Measure of Association

The measure of association is used to determine statistical dependency among the lexical features considered for the labeling of clusters. It is useful in selecting only those n-grams which are statistically important and aids in eliminating independent features. An n-gram is a contiguous sequence of n words from a textual string. The various measure of association supported in the systems are:

- ll - log likelihood ratio (default measure) [10] [15]
- pmi - point-wise mutual information [8] [11]
- tmi - true mutual information [15]
- x2 - chi-squared test [17]
- phi - phi coefficient [6] [7]
- tscore - t-score [7]
- Dice - Dice coefficient [23]



- odds - odds ratio [3]
- leftFisher - left Fishers test [18]
- rightFisher - right Fishers test [18]

### **2.2.5 Score Cutoff**

After applying a measure of association, we look for a threshold value above which features are considered interesting for our labeling mechanism. The terms having degrees of association values less than this threshold can be ignored without worrying about their impact on the final result. (In general, a user should set this value based on his or her past experience with the process.) The default value for this parameter is 10.

### **2.2.6 Feature Selection**

Another parameter that affects the overall clustering method is feature selection. The options available for feature selection are unigram, bigram, co-occurrence and target co-occurrence. Unigram and bigram are variations of n-grams; they represent respectively a single word or a pair of words as the features. The co-occurrence features are similar to bigrams except the order of words in the pair becomes irrelevant. For target co-occurrence, unordered pairs of words are selected, in which one word is always a word that is specified by user as the concerned word.

### **2.2.7 Cluster and Labeling Examples**

In this section, we illustrate the labeling mechanism of SenseClusters by using three examples. We are concerned primarily with how the following parameters influence the labeling of a cluster:

- label stop: the file containing stop words
- label remove: the frequency cutoff for feature selection
- label window: the window size for feature selection

- label stat: the method of association for features
- label rank: the threshold score below which all features will be discarded

To illustrate the labeling process, we use a dataset that contains information about Bill Clinton, Tony Blair and Ehud Barack. We expect the SenseClusters labeling system to produce these topics (or information closely related to these topics) as cluster labels. Additionally, the information associated with the labels should only be present in the corresponding cluster. For example, if the labeling mechanism were totally accurate, we would expect results similar to the following:

*Cluster 0: (Descriptive): Ehud Barack, Israeli politician, Prime Minister, Labour Party, Defense Minister, Stanford University, Ariel Sharon, Yassir Arafat*

*Cluster 0:(Discriminating):Ehud Barack, Israeli politician, Defense Minister, Stanford University, Ariel Sharon*

*Cluster 1 (Descriptive): Bill Clinton, American politician, US President, Dont ask, dont tell, Clinton Foundation, Yale School, Arkansas Governor, budget surplus, mass destruction*

*Cluster 1 (Discriminating): Bill Clinton, American politician, US President, Dont ask, dont tell, Clinton Foundation, Yale School, Arkansas Governor, budget surplus*

*Cluster 2 (Descriptive): Tony Blair, Labour Party, Prime Minister, United Kingdom, New Labour, Oxford 1975, Iraq War, 2001 invasion, War Terror, mass destruction, Yassir Arafat*

*Cluster 2 (Discriminating): Tony Blair, United Kingdom, New Labour, Oxford 1975, Iraq War, 2001 invasion, War Terror, mass destruction*

Example 1:

For the first example, we use the default values of the parameters (given in 2.2.7) and provide data only for those parameters having no default values. In this case, the input command is as follows:

```
$discriminate.pl "TonyBillEhud1_-test.xml" --target "target.regex"
  --token "token.regex" --feature bi --format f16.06 --remove 5
  --stat ll --context o2 --cluststop pk2 --space vector --sim cos
  --clmethod rb --crfun i2 --label_remove 5 --label_stat ll
  --label_rank 10 --prefix "TonyBillEhud1_"
```

SenseClusters produced the following clusters and labels for this set of parametric values..

*Cluster 0 (Descriptive): Yasser Arafat, Middle East, York Times, British Prime, Camp David, Israeli Prime, Minister <head>BTE</head>, U S, Prime Minister, New York*

*Cluster 0 (Discriminating): Yasser Arafat, Middle East, British Prime, Camp David, Israeli Prime, Minister <head>BTE</head>, Prime Minister*

*Cluster 1 (Descriptive): Al Gore, prime minister, White House, W Bush, George W, York Times, President <head>BTE</head>, U S, of the, New York*

*Cluster 1 (Discriminating): Al Gore, prime minister, White House, W Bush, George W, President <head>BTE</head>, of the*

In this example, BTE represents a placeholder in the data representing the names Bill Clinton, Tony Blair, or Ehud Barack.

The clustering algorithm produced only two clusters and we were expecting three. Obviously if the clustering algorithm does not generate a good cluster space then we cannot expect good labels. This causes an infusion of topics as labels. Also, since stop words were ignored, we have words in the labels which are obscuring the actual meaning of the cluster. These results can be verified at:

<http://marimba.d.umn.edu/SChtdocs/user1335887624/user.clusterlabels.pk2>

Example 2:

For the second example, we provide values for all the labeling parameters and change the feature selection process from bigram to target co-occurrence. The corresponding input command is as follows:

```
$discriminate.pl "TonyBillEhud3_-test.xml" --target "target.regex"
--token "token.regex" --format f16.06 --feature coc --remove 10
--stop stopfile --window 4 --stat rightFisher --stat_rank 10
--context o2 --clusters 3 --space vector --sim corr --crfun h2
--clmethod rbr --label_stop label_stopfile --label_remove 10
--label_window 4 --label_stat rightFisher --label_rank 10
```

--prefix "TonyBillEhud3\_"

The clusters and labels corresponding to this case are shown below. Note that what appears here as a single term represents the co-occurrence of the term with a blank string.

*Cluster 0 (Descriptive):* POLITICS Minister, <head>B.T.E</head> Prime, Minister Arafat, Bush <head>B.T.E</head>, London <head>B.T.E</head>, - <head>B.T.E</head>, President <head>B.T.E</head>, peace Prime, <head>B.T.E</head> Minister, ISRAEL Minister, Syrian Minister, Israel <head>B.T.E</head>, Monday <head>B.T.E</head>, Prime government, Minister leader

*Cluster 0 (Discriminating):* POLITICS Minister, <head>B.T.E</head> Prime, Minister Arafat, Bush <head>B.T.E</head>, London <head>B.T.E</head>, - <head>B.T.E</head>, President <head>B.T.E</head>, peace Prime, <head>B.T.E</head> Minister, ISRAEL Minister, Syrian Minister, Israel <head>B.T.E</head>, Monday <head>B.T.E</head>, Prime government, Minister leader

*Cluster 1 (Descriptive):* Jimmy <head>B.T.E</head>, same <head>B.T.E</head>, Al <head>B.T.E</head>, Bill <head>B.T.E</head>, White <head>B.T.E</head>, <head>B.T.E</head> Clinton, Bob <head>B.T.E</head>, Democratic <head>B.T.E</head>, <head>B.T.E</head> minister; Ronald <head>B.T.E</head>, John <head>B.T.E</head>

*Cluster 1 (Discriminating):* Jimmy <head>B.T.E</head>, same <head>B.T.E</head>, Al <head>B.T.E</head>, Bill <head>B.T.E</head>, <head>B.T.E</head> Clinton, Bob <head>B.T.E</head>, Democratic <head>B.T.E</head>, <head>B.T.E</head> minister; Ronald <head>B.T.E</head>, John <head>B.T.E</head>

*Cluster 2 (Descriptive):* Bush -, <head>B.T.E</head> House, George <head>B.T.E</head>, White <head>B.T.E</head>, - Arafat, York -, Yasser <head>B.T.E</head>, - House, York <head>B.T.E</head>, CLINTON <head>B.T.E</head>, Clinton -, category -, WASHINGTON <head>B.T.E</head>, House -, <head>B.T.E</head> -, 2000 -, President -, President President

*Cluster 2 (Discriminating):* Bush -, <head>B.T.E</head> House, George <head>B.T.E</head>, - Arafat, York -, Yasser <head>B.T.E</head>, - House, York <head>B.T.E</head>, CLINTON

*<head>B.T.E</head>, Clinton →, category →, WASHINGTON <head>B.T.E</head>, House →,  
<head>B.T.E</head> →, 2000 →, President →, President President*

None of these labels provide any clear indication of the specific topic of the query (that is, Clinton, Blair, Barack) although they do include some features that are associated with them. The labels are verbose and include repetitions of unreadable terms even though the stop list is used. As a result the labeling mechanism does not provide the user with a clear indication of the meaning of the clusters. Labels for this case can be obtained from:

[http://marimba.d.umn.edu/SC-htdocs/TonyBillEhud3\\_1358633481/](http://marimba.d.umn.edu/SC-htdocs/TonyBillEhud3_1358633481/)

Example 3:

Most of the values of the parameters in this example are similar to Example 1 except we specify that 3 clusters be formed. This eliminates the issue of inappropriate cluster formation and permits more focus on the labeling process. The input command is shown below.

```
$discriminate.pl "TnyBilEud4_-test.xml" --format f16.06 --window 5  
--token "token.regex" --target "target.regex" --feature coc  
--remove 10 --stat tscore --stat_rank 10 --stop stopfile  
--context o2 --clusters 3 --space similarity --clmethod rbr  
--crfun h2 --label_stop label_stopfile --label_stat tscore  
--sim cos --label_window 10 --label_remove 10 --label_rank 10  
--prefix "TnyBilEud4_"
```

Labels obtained for the above command from SenseClusters are:

*Cluster 0 (Descriptive): Yasser Arafat, Prime <head>B.T.E</head>, York Times, British Prime, Camp David, British Minister, Minister <head>B.T.E</head>, Israeli Prime, President Clinton, Prime Minister*

*Cluster 0 (Discriminating): Prime <head>B.T.E</head>, British Prime, British Minister, Minister <head>B.T.E</head>, Israeli Prime, President Clinton, Prime Minister*

*Cluster 1 (Descriptive): George Bush, Bill Clinton, Al Gore, White House, former <head>B.T.E</head>, Washington →, York Times, President <head>B.T.E</head>, COLUMN →, former President*

*Cluster 1 (Discriminating): George Bush, Al Gore, White House, former <head>B.T.E</head>, President <head>B.T.E</head>, former President*

*Cluster 2 (Descriptive): Yasser Arafat, Bill Clinton, prime minister, Washington -, Camp David, minister <head>B.T.E</head>, Ariel Sharon, United States, PrimeMinister <head>B.T.E</head>, COLUMN -*

*Cluster 2 (Discriminating): prime minister, minister <head>B.T.E</head>, Ariel Sharon, United States, PrimeMinister <head>B.T.E</head>*

If we closely observe these labels, we conclude that the label for Cluster 0 is not clear and it shows evidence for both topics, Tony Blair and Ehud Barack. Conversely, labels for Cluster 2 do not provide any strong evidence for any of these three topics. However, Cluster 1 provides reliable evidence to conclude that labels provide clear information about former president Bill Clinton. Results for this test case can be verified at:

[http://marimba.d.umn.edu/SCHtdocs/Test21335986679/Test2.cluster labels.](http://marimba.d.umn.edu/SCHtdocs/Test21335986679/Test2.cluster%20labels)

### **2.2.8 Summary**

The examples shown in the previous section illustrate some of the difficulties inherent in the cluster labeling process. The major problem with these labels is the presence of extraneous terms which do not contribute in identifying the central theme of the cluster. Thus we investigate using tf-idf to improve the quality of the terms in the labels. The details of the tf-idf experiments are given in chapter 5.

Another problem which occurs in almost all labels is an inability to capture important signals because of small size of n-gram. For example, if we can capture the term British Prime Minister instead of British Prime, British Minister, we will have better information. In Chapter 4, we provide details of the impact of different values of n in NGRAM in cluster labeling.

## **2.3 Evaluating Cluster Labels**

Cluster labeling is an area of research that has been on going for several years. However, very little work has been done in the area of label evaluation - the mechanism of evaluating the quality of labels assigned to the clusters. The most widely used means of evaluating labels of a cluster is to compare them with gold standard keys [14] [4] [20]. An expert examines the contents of each cluster and identifies words that best define the cluster. These words are referred to as the gold standard key or gold key for the cluster. The labels generated by the clustering system are compared with the gold keys to determine the performance of the labeling process.

We have developed a system that automates the evaluation process and assigns an accuracy score to the labeling process. The system reduces manual input and effort on the part of the user. The labeling evaluation process is not subjective but rather employs a statistical means of determining the similarity between labels and gold keys. In the next chapter we define and test such a mechanism.

### **3 A Cluster Label Evaluation System**

Our goal is to build an automatic system that evaluates cluster labels - a system that is flexible, extensible and reliable. To support flexibility, the system provides options that allow the user to exercise control over the evaluation process. It is extensible in that it allows the user to add to or modify the existing mechanisms. With respect to reliability, the system utilizes similarity procedures [19] and the Hungarian algorithm [16] to assign an accuracy score to the labeling process. Overall, the system requires minimal input and user effort.

To evaluate the labels, the user specifies topics that are known to have relevant data contained within the corpus. These topics are referred to as gold standard topics for the collection. Ideally, SenseClusters would produce clusters which only contain data associated with a single gold topic, and the cluster labels generated by SenseClusters would reflect this. Data associated with a gold standard topic represent a gold standard key. The label evaluation system compares the cluster labels with the gold standard keys, and the more they agree with the gold keys, the better the labeling process is considered to be. The user may provide the gold standard key or specify the source from which the gold standard key can be obtained. We currently use Wikipedia [4] as the source. However, new sources can easily be added. We select Wikipedia because it provides free, open and vast amounts of data on most topics. Another source which may be included in the future is Wordnet [2].

#### **3.1 User Options**

Before defining the label evaluation algorithm, we specify the set of user options that control the evaluation process.

##### **3.1.1 SenseClusterLabelFileName**

This parameter specifies the name of the file that contains the labels for the clusters generated by SenseClusters. The format of the file is the same as that generated by SenseClusters. As an example, suppose we want to evaluate the labels of 3 clusters generated by SenseClusters. The



possible contents of the file are listed below.

*Cluster 0 (Descriptive): George Bush, Russian President, British Prime, British Minister, India Pakistan, US George, Prime Minister*

*Cluster 0 (Discriminating): Russian President, British Minister, India Pakistan, US George*

*Cluster 1 (Descriptive): George Bush, British Prime, weapons mass, United Nations, September 11, mass destruction, United States, Prime Minister, military action*

*Cluster 1 (Discriminating): United Nations, September 11, United States*

*Cluster 2 (Descriptive): George Bush, weapons destruction, prime minister, axis evil, Saddam Hussein, weapons mass, mass destruction, Gulf War, military action, Iraqi leader*

*Cluster 2 (Discriminating): weapons destruction, prime minister, axis evil, Saddam Hussein, Gulf War, Iraqi leader*

### **3.1.2 GoldKeyFileName**

The GoldKeyFileName is used to establish the gold standard topics for the clusters, the mapping between the clusters and gold topics, and the gold standard keys for the clusters. The user passes two parameters to the label evaluation system through this file, LabelComparisonMethod and GoldKeyDataSource.

The LabelComparisonMethod specifies how the mapping information between gold topics and clusters will be provided. The user can elect to provide the mapping information directly or let the system find the best match. In either case, the user himself specifies the gold standard topics by designating the option as follows:

1. 'direct' - user will provide the mapping information
2. 'automate' - the system will determine the best possible mapping between cluster's label and gold topics.

The user may provide the gold standard keys for each cluster or specify the source from which the gold standard keys can be accessed. Recall that a clusters gold standard key is data associated with a gold standard topic. The GoldKeyDataSource specifies the option chosen by the user and is identified as follows:

1. 'wikipedia' - fetch data for gold topics from Wikipedia
2. 'userData' - user provides the gold standard keys for each cluster.

### **3.1.3 WeightRatio**

WeightRatio is used to specify the significance of the discriminating labels over descriptive labels in the SenseCluster cluster label file. Since discriminating labels are unique to a given cluster, its similarity score with the gold keys should have more weight than that of the descriptive labels. The user may specify a value; in general, the value should be equal to or greater than 1. The default value for this parameter is 10.

### **3.1.4 IsClean**

The IsClean flag specifies whether the temporary file that the system created during the evaluation process should be retained once the process is completed. This flag provides the user with an option to evaluate the programs functionality more closely. The temporary file contains data fetched from Wikipedia (or other source). The default value for IsClean is 0, an indication that the file is to be deleted upon completion of the labeling process.

The label evaluation system is implemented in Perl and released as open source code in CPAN. The user provides the values of the parameters through the hash object. If the user elects to use the default value of a parameter, then the parameter is not included as part of the input. . A sample input for the program is illustrated in Figure 1.

```

# Calling the LabelEvaluation modules by passing the following options
%inputOptions = (
    SenseClusterLabelFileName => 'labelFile.txt',
    LabelComparisonMethod => 'autcmate',
    GoldKeyFileName => 'goldKeyFile.txt',
    GoldKeyLength => 2000,
    GoldKeyDataSource => 'Wikipedia',
    WeightRatio => 10,
    StopListFileLocation => 'stoplist.txt',
    IsClean => 1,
    Verbose => 1,
    Help => 0
);

```

Figure 1: A Sample Input Passed to the LabelEvaluation Module

### 3.2 Label Evaluation

Figure 2 contains a flowchart for the cluster label evaluation system. In this section we provide details of the steps involved in the evaluation process, including calculating the similarity between the cluster labels and gold standard keys and calculating the accuracy of the label assignments.

As previously noted, obtaining the gold keys can be done in one of two ways: (1) the user provides a data file for the gold topics that the system uses for similarity calculations, or (2) the user specifies an external source from which the system will access data on the gold topics. The external source is Wikipedia [12]. For fetching the Wikipedia data, we use the WWW::Wikipedia module from the CPAN [1].

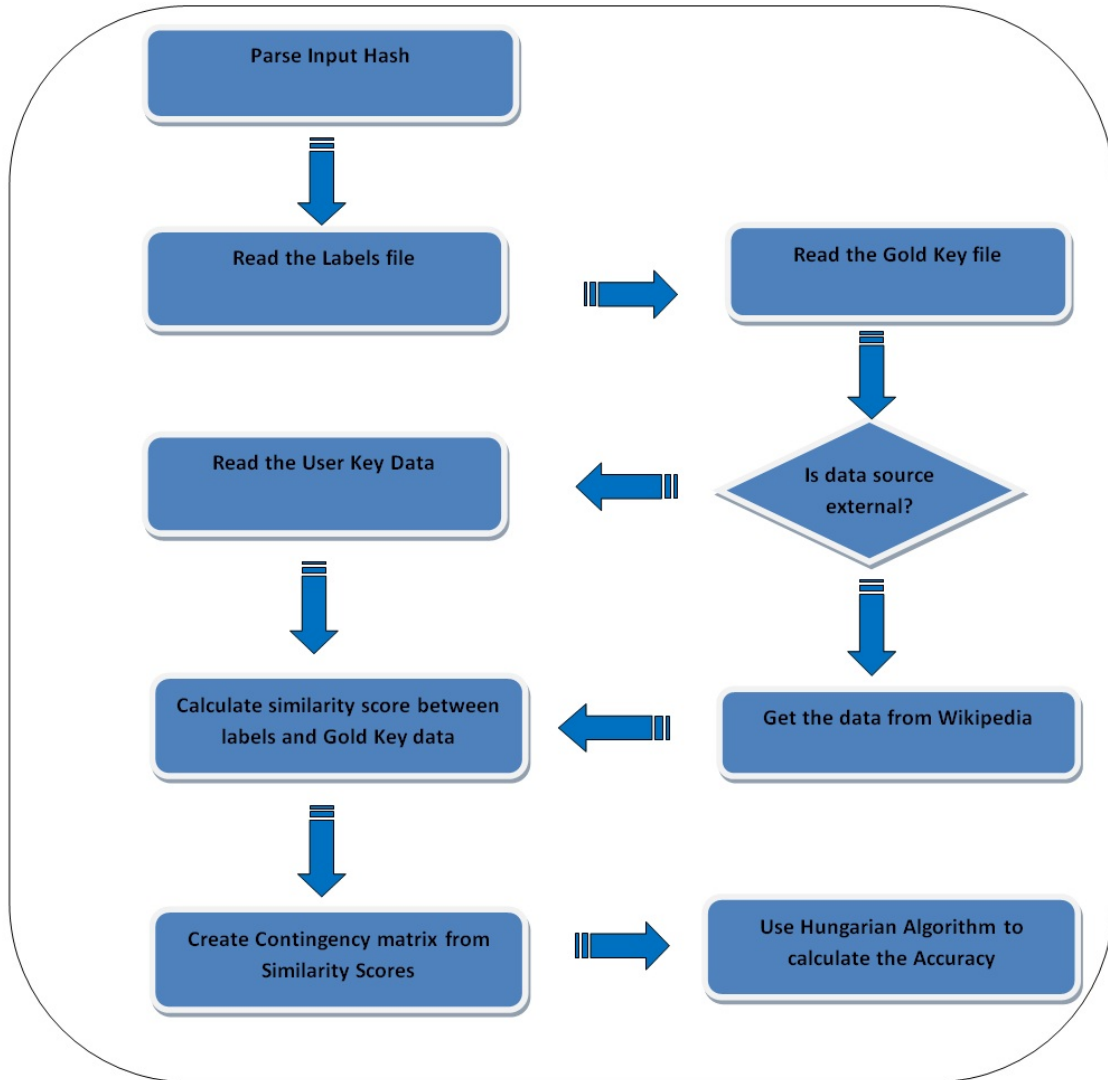


Figure 2: Flowchart for Label Evaluation System

### 3.2.1 User provides the mapping information

When specifying the name of the cluster label file, the user has the option of providing the mapping of the clusters to the gold standard topics. The steps below correspond to this option.

1. Read the clusters and their labels - descriptive and discriminating - from the cluster label file.
2. Read the gold key file.

3. Access the gold standard topics and the mapping information. (In this case, we assume the mapping option information is “direct.”)
4. Obtain the additional data for the gold topics.
 

Case A: If the user has elected to provide data for the gold topics, read the gold standard keys from the file provided by the user.

Case B: If the user does not provide the gold topic data, the system obtains the gold standard keys from Wikipedia. (See figure 3.)
5. Create a contingency matrix with the similarity scores of a cluster’s label against the gold standard keys (obtained in Step 4).
6. Using the mapping provided by the user (Step 3), calculate the diagonal score for the contingency matrix.
7. Calculate the overall accuracy score for the clusters label assignment.

```
#The following code snippet will show how to use this module.

# Including the LabelEvaluation Module.
use Text::SenseClusters::LabelEvaluation::Wikipedia::GetWikiData;

# Defining the topic name for which we will create the file containing their detail
# data from the wikipedia.
my $topicName = "BillClinton";

# The following code will call the getWikiDataForTopic() function from the
# GetWikiData modules. It will create the file containing the wikipedia
# information about the topic.
my $fileName =
    Text::SenseClusters::LabelEvaluation::Wikipedia::GetWikiData::getWikiDataForTopic(
        $topicName);

print "\nName of the File created for the topic \'$topicName\' is $fileName \n";
```

Figure 3: Module to Access Data from Wikipedia

### 3.2.2 User does not provide the mapping information

If the user does not provide the mapping information between the clusters and gold standard topics, we use the Hungarian algorithm [16] to compute the mapping, shown below.

1. Read the clusters and their labels - descriptive and discriminating - from the cluster label file.
2. Read the gold key file.
3. Access the gold standard topics. (In this case, we assume the mapping option is 'automate'.)
4. Obtain the additional data for the gold topics.  

Case A: If the user elected to provide data for the gold topics, read the gold standard keys from the file provided by the user.

Case B: If the user does not provide the gold topic data, the system obtains data the gold standard keys from Wikipedia.
5. Create a contingency matrix with the similarity scores of a cluster's label against the gold standard keys (obtained in Step 4).
6. Use the Hungarian algorithm (Section 3.4) to determine the mapping of the clusters with the gold topics.
7. Using this mapping, calculate the total diagonal score for the contingency matrix.
8. Calculate the overall accuracy score for the clusters label assignment.

### **3.2.3 Calculating the Similarity Score**

The similarity score between the cluster labels and the gold standard keys are calculated using the Text::Similarity module [19] from CPAN. The Text::Similarity takes two strings and a file containing the stop words as the inputs and returns the following overlapping scores:

- (a) Lesk score
- (b) raw Lesk score
- (c) precision
- (d) recall
- (e) F score

- (f) Dice score
- (g) E score
- (h) cosine score

In general, the raw-Lesk [19] score provides the best results as it produces the pure overlapping score. This is the default scoring mechanism. As an example of its use, consider two strings: (1) “*India is the world’s most populous democracy*” and (2) “*Report from India says, the most populous democracy will go for election in the summer*”. The largest matching string with a word count of 3 is “*most populous democracy*” and it occurs once. The other matching words are ‘*India*’ and ‘*the*’. The word ‘*the*’ is ignored since it is contained in the stop word list. So, the raw- Lesk score is 10:  $(3^2 + 1^2)$ . Figure 4 contains a module for calculating the similarity score. For more detail, information, see the Text::Similarity module [19].

### 3.3 Hungarian Algorithm

Once we calculate the similarity scores between the labels of the clusters and the gold standard keys, we create the contingency matrix from these scores. The contingency matrix provides a way to express and visualize these similarity scores in a condensed way. The contingency matrix is then passed as input to the Hungarian Algorithm [16]. The Hungarian algorithm tries to maximize the diagonal scores and hence provide the best match of cluster labels with the gold topics. Figure 5 illustrates this process. We have used the module Algorithm::Munkres for the Hungarian calculation [16].

The original contingency matrix appears at the top of Figure 5. The individual value of the matrix shows the similarity scores between a cluster and a gold standard topic. For example, 55 is the similarity score between the Cluster0s label and Bill Clintons gold keys. The second contingency matrix is obtained when we apply the Hungarian algorithm to the original contingency matrix. The Hungarian algorithm tries to maximize the diagonal values of the contingency matrix. For example, the possible diagonal values for the original contingency matrix are {55, 11, 143, sum=209}, {55, 66, 11, sum=132}, {231, 89, 143, sum=463}, {231, 66, 188, sum=435}, {242, 11, 188, sum=441} and {242, 89, 11, sum=342}. The combination of diagonal values that represent the maximum

```

# The following code snippet will show how to use SimilarityScore.
package Text::SenseClusters::LabelEvaluation::Test_SimilarityScore;

# Including the LabelEvaluation Module.
use Text::SenseClusters::LabelEvaluation::SimilarityScore;

my $firstString = "IBM:: vice president, million dollars, Wall Street, Deep Blue, ".
    "International Business, Business Machines, International Machines, ".
    "United States, Justice Department, personal computers";

my $secondString = "vice president, million dollars, Deep Blue, International ".
    "Business, Business Machines, International Machines, United".
    " States, Justice Department";

my $similarityObject = Text::SenseClusters::LabelEvaluation::SimilarityScore->
    new($firstString,$secondString, "./stoplist.txt");

print "Score:: $score \n";
print "Lesk Score :: $allScores{'lesk'} \n";
print "Raw Lesk Score :: $allScores{'raw_lesk'} \n";
print "precision Score :: $allScores{'precision'} \n";
print "recall Score :: $allScores{'recall'} \n";
print "F Score :: $allScores{'F'} \n";
print "dice Score :: $allScores{'dice'} \n";
print "E Score :: $allScores{'E'} \n";
print "cosine Score :: $allScores{'cosine'} \n";
print "\n\n";

```

Figure 4: Module to Calculate the Similarity Score

summation is {231, 89 and 143}. The second contingency matrix created above reflects these as its diagonal elements.

Finally, the mapping between clusters and gold standard topics is established using these diagonal elements. So, our mapping in Figure 5 shows Cluster0 with Ehud Barack, Cluster1 with Bill Clinton and Cluster2 with Tony Blair.

The accuracy of the label assignment is calculated as follows:

$$Accuracy = \left( \frac{Sum(Diagonal\ scores\ of\ Hungarian\ contingency\ matrix)}{Sum(All\ the\ scores\ of\ Hungarian\ contingency\ matrix)} \right) \quad (1)$$



Original Contingency Matrix:

	Bill Clinton	Ehud Barak	Tony Blair
Cluster0	55	231	242
Cluster1	89	11	66
Cluster2	188	11	143

Contingency Matrix after Hungarian Algorithm (Maximizing the diagonal scores):

	Ehud Barak	Bill Clinton	Tony Blair
Cluster0	231	55	242
Cluster1	11	89	66
Cluster2	11	188	143

Final Conclusion using Hungarian Algorithm::

Cluster0	<-->	Ehud Barak
Cluster1	<-->	Bill Clinton
Cluster2	<-->	Tony Blair

Figure 5: Contingency Matrix and Hungarian Algorithm Result

## 4 Evaluation of the Algorithm

In this chapter we describe four experiments performed using the cluster label evaluation algorithm.

### 4.1 Test Data

These experiments use data created by Kulkarni [12] that is based on the New York Times (January 2000 to June 2002) corpus. After gathering articles on a set of topics, she conflated the topics with a pseudo word. For example, articles on Tony Blair, Vladimir Putin and Saddam Hussein were collected and all the instances of these names were replaced with a conflated word T\_V\_S. Using this data, we then generate clusters and their labels using SenseClusters and the cluster labels are compared with the gold standard keys using our label evaluation system. The accuracy of the label assignment is calculated.

### 4.2 Experiments

We performed four experiments using the LabelEvaluation module. Each experiment is based on a different set of user input options (described in Section 3.1). The four cases are identified below and their test results presented. In each case the user specifies the gold standard topics namely, Tony Blair, Vladimir Putin and Saddam Hussein. The user may or may not provide the mapping of the clusters to the gold topics. Additionally, the gold standard keys may be provided by the user or obtained using Wikipedia. Combinations of these options lead to the four cases.

#### 4.2.1 Case 1: User provides the mapping information and the gold standard keys

The value direct for the LabelComparisonMethod parameter means that user will provide the mapping between the labels of the clusters and the gold standard topics. Furthermore, the value userData for the GoldKeyDataSource parameter signifies that the user will provide the gold standard keys (that is, relevant data about the gold topics). The gold keys may consist of a short description of the topics or perhaps more detailed information on each of the topics.

Suppose the following information is provided for this case. First is the users assessment of the best mapping between each cluster and the gold standard topics. The mapping is then followed by a short description of each of the gold topics; these descriptions represent the gold standard keys.

*Cluster0:::Tony Blair*

*Cluster1:::Vladimir Putin*

*Cluster2:::Saddam Hussein*

*Tony Blair::: Anthony Charles Lynton Blair (born 6 May 1953) is a British Labour Party politician who served as the Prime Minister of the United Kingdom from 1997 to 2007. He was the Member of Parliament (MP) for Sedge field from 1983 to 2007 and Leader of the Labour Party from 1994 to 2007. He resigned from all of these positions in June 2007.*

*Vladimir Putin::: Vladimir Vladimirovich Putin (born 7 October 1952) is a Russian politician who has been the President of Russia since 7 May 2012. Putin previously served as President from 2000 to 2008, and as Prime Minister of Russia from 1999 to 2000 and again from 2008 to 2012. Putin was also previously the Chairman of United Russia.*

*Saddam Hussein::: Saddam Hussein Abd al-Majid al-Tikriti 28 April 1937, 30 December 2006, was the fifth President of Iraq, serving in this capacity from 16 July 1979 until 9 April 2003. A leading member of the revolutionary Arab Socialist Ba'ath Party.*

The results of the cluster label evaluation are shown in Table 1. The table shows the contingency matrix which is created based on the user-provided mapping of clusters with the gold standard topics. Based on this contingency matrix we observe that our algorithm provides similarity scores which match with the expectation of a human expert. Our system provides higher similarity scores for those clusters and gold standard keys which are judged as the correct mapping by an expert.

Table 1: Case 1 User Provides the Mapping Information and the Gold Standard Keys

	Contingency Matrix based on User Mapping			Mapping provided by user
	Saddam Hussein	Vladimir Putin	Tony Blair	Results
Cluster 0	72	61	74	Cluster2 - Saddam Hussein Cluster1 - Vladimir Putin Cluster0 - Tony Blair
Cluster 1	17	29	22	
Cluster 2	173	45	70	
Accuracy	49.02%			

#### 4.2.2 Case 2: User provides mapping information and Wikipedia is used to generate the gold keys

The second case is similar to the first except that the system generates the gold keys using Wikipedia rather than the data being user-provided. If the topic under consideration is not present in Wikipedia, the description provided will be empty, resulting in a similarity score of zero.

The users assessment of the best mapping between each cluster and the topic of the gold keys is the same as Case 1:

*Cluster0::Tony Blair*

*Cluster1::Vladimir Putin*

*Cluster2::Saddam Hussein*

The results of the cluster label evaluation are shown in Table 2. The contingency matrix of this case is quite similar to the previous case. It is also based on user-provided mapping information. Similarity scores of clusters and gold standard keys are comparable to the previous case. Like the last case, it also indicates that similarity scores between clusters and gold standard topics is in accordance with the mapping provided by expert.

#### 4.2.3 Case 3: System establishes mapping information and the user provides the gold keys

In the third case, the parameter LabelComparisonMethod is automate which means that the user does not want to suggest the mapping between the clusters labels and gold standard topics. However,

Table 2: Case 2 User Provides Mapping Information and Wikipedia is Used to Generate the Gold Key

	Contingency Matrix based on User Mapping			Mapping provided by user
	Saddam Hussein	Vladimir Putin	Tony Blair	Results
Cluster0	89	76	88	Cluster2 - Saddam Hussein
Cluster1	155	110	144	Cluster1 - Vladimir Putin
Cluster2	222	71	70	Cluster0 - Tony Blair
Accuracy	40.98%			

the user will provide relevant data about the gold topics. Since the mapping is not specified, the system will use the similarity scores between the cluster labels and gold keys and attempt to align the clusters with the gold topics. The alignment is done using the Hungarian algorithm, which tries to maximize the accuracy of the alignment.

The gold keys are the same as those in Case 1:

*Tony Blair::: Anthony Charles Lynton Blair (born 6 May 1953) is a British Labour Party politician who served as the Prime Minister of the United Kingdom from 1997 to 2007. He was the Member of Parliament (MP) for Sedge field from 1983 to 2007 and Leader of the Labour Party from 1994 to 2007. He resigned from all of these positions in June 2007.*

*Vladimir Putin::: Vladimir Vladimirovich Putin (born 7 October 1952) is a Russian politician who has been the President of Russia since 7 May 2012. Putin previously served as President from 2000 to 2008, and as Prime Minister of Russia from 1999 to 2000 and again from 2008 to 2012. Putin was also previously the Chairman of United Russia.*

*Saddam Hussein::: Saddam Hussein Abd al-Majid al-Tikriti 28 April 1937, 30 December 2006, was the fifth President of Iraq, serving in this capacity from 16 July 1979 until 9 April 2003. A leading member of the revolutionary Arab Socialist Ba'ath Party.*

The results of the cluster label evaluation for this case are shown in Table 3. Here we show the raw contingency matrix and the similarity score between the clusters and gold standard topics. We also show the contingency matrix with the correct mapping between the cluster and topic using the Hungarian algorithm. We conclude that our mapping algorithm from cluster to gold topic is reliable.

The result in this case matches the result of case 1, where a human expert has provided the mapping information.

Table 3: Case 3 System Establishes Mapping Information and the User Provides the Gold Keys

	Contingency Matrix with similarity scores			Contingency Matrix after Hungarian Algorithm		
	Saddam Hussein	Vladimir Putin	Tony Blair	Tony Blair	Vladimir Putin	Saddam Hussein
Cluster 0	72	61	74	74	61	72
Cluster 1	17	29	22	22	29	17
Cluster 2	173	45	70	70	45	173
Accuracy	49.02%			Results	Cluster0 - Tony Blair Cluster1 - Vladimir Putin Cluster2 - Saddam Hussein	

#### 4.2.4 Case 4: System establishes mapping information and Wikipedia is used for the gold keys

In Case 4 the user specifies only the gold topics: *Tony Blair*, *Vladimir Putin*, *Saddam Hussein*. The mapping between the clusters labels and gold topics and the relevant data about the gold topics are established by the system. Wikipedia is used to generate the gold keys. This case shows the use of system with minimal user input.

The results of the cluster label evaluation for this case are shown in Table 4. The result of this case is similar to that of the case 2, but loss of supporting data clearly weaken the results. If the original clusters were cleaner i.e. more tightly bounded together, it is a open question if the same result will occur.

### 4.3 Summary

We have performed various experiments using different sets of data and configurations. As we move to various test cases, manual involvement decreases. However, the system maintains the comparable accuracy with similar results as long as either the mapping or the gold standard keys are provided. The first case represents the traditional approach where a human expert is responsible for providing all the information. For the last test case, where only gold topics are provided, results are no longer

Table 4: Case 4 System Establishes Mapping Information and Wikipedia is Used for the Gold Keys

	Contingency Matrix with similarity scores			Contingency Matrix after Hungarian Algorithm		
	Saddam Hussein	Vladimir Putin	Tony Blair	Vladimir Putin	Tony Blair	Saddam Hussein
Cluster 0	89	76	88	76	88	89
Cluster 1	155	110	144	110	144	155
Cluster 2	222	71	70	71	70	222
Accuracy	43.12%			Results	Cluster0 - Vladimir Putin Cluster1- Tony Blair Cluster2 - Saddam Hussein	

compatible with cases 1, 2 and 3.

## 5 N-gram Features Selection

### 5.1 Description

Most systems generate cluster labels using either unigrams or bigrams for feature selection. We want to determine the effect of a larger value of  $n$  for the  $n$ -gram [24] in label selection. In this chapter, we use both bigrams and trigrams for feature selection in the various experiments. The motivation behind using a greater value of  $n$  is to capture information that we miss when using a smaller value of  $n$  while doing feature selection. For instance, if we consider an entity, '*New York Times*', or a person, '*Prime Minister Tony Blair*', the information captured when using unigrams is insignificant. Unigram features for '*New York Times*' are '*New*', '*York*' and '*Times*'. Some individual words may be stop words and if so will be ignored by the label selection process. However, together they become an important indicator of content. Higher-level  $n$ -grams are also useful in capturing the phrases which may otherwise be ignored. For example, '*the mango people*' is widely used to refer to the common man of India. However, if using the unigram for feature selection, then the feature list produces '*the*', '*mango*' and '*people*', which individually have meanings quite different from actual meaning of the phrase. Many strong signals are lost if we use smaller values of  $n$  for  $n$ -gram feature selection.

The  $n$ -gram feature [24] for a dataset is selected by considering  $n$  consecutive words as part of a single feature. For example, Table 5 shows the unigram, bigram, trigram, and quadragram feature selection for this sentence: '*Bill Clinton is an American politician who served from 1993 to 2001 as the 42nd President of the United States.*'

We observe from Table 5 that some of the important signals like '*1993 to 2001*' are missed if we are using either a unigram or bigram feature set. Such signals give a strong indication about the topic, which in this case is '*Bill Clinton*'.



Table 5: Example Showing Unigram, Bigram, Trigram, Quadragram Feature Selection

<i>Unigram</i>	<i>Bigram</i>	<i>Trigram</i>	<i>Quadragram</i>
bill, clinton, is , as, American, politician, who, served, from , 1993, to, 2001, the, 42 <sup>nd</sup> , president, of, united, states	bill clinton, clinton is, is an, an american, american politician, politician who, who served, served from, from 1993, 1993 to, to 2001, 2001 as, as the, the 42 <sup>nd</sup> , 42nd president, president of, of the, the united, united states	bill clinton is, clinton is an, is an american, an american politician, american politician who, politician who served, who served from, served from 1993, from 1993 to, 1993 to 2001, to 2001 as, 2001 as the, as the 42 <sup>nd</sup> , the 42nd president, 42nd president of, president of the, of the united, the united states	bill clinton is an, clinton is an american, is an American politician, an american politician who, american politician who served, politician who served from, who served from 1993, served from 1993 to, from 1993 to 2001, 1993 to 2001 as, to 2001 as the, 2001 as the 42nd, as the 42 <sup>nd</sup> president, the 42nd president of, 42nd president of the, president of the united, of the united states

## 5.2 Experimental Setup

### 5.2.1 Assumption

All experiments in this section assume that clustering of the corpus is important, a key and influential factor in identifying the topic. If the clustering process produces poor clusters, then identifying the correct topics for these clusters is a difficult process that will produce a low accuracy result.

### 5.2.2 Training and testing dataset

Our experiments are based on the corpus created by Kulkarni [12]. It contains various New York Times articles on former American President Bill Clinton, former British Prime Minister Tony Blair and former Israeli Prime Minister Ehud Barak. The dataset has approximately the same number of articles on each subject. It also contains articles which in general relate to them with issues relevant to the era when they were active as the heads of their respective countries. Thus, the dataset represents a real world environment where topics are intertwined and do not have clear boundaries.

Figures 6 and 7 give the code snippets for bigram and trigram selection and frequency counts.

```
# Iterating through each Words.
foreach my $word (@wordArray){
    $word = lc($word);
    if($previousWord eq ""){
        $previousWord = $word;
        next;
    }
    $bigramToken = $previousWord." ".$word;
    if($hashWordDocumentCount{$bigramToken}{$documentName}){
        $hashWordDocumentCount{$bigramToken}{$documentName} =
            $hashWordDocumentCount{$bigramToken}{$documentName} + 1;
        if($maxWordFrequencyOfDocument <
            $hashWordDocumentCount{$bigramToken}{$documentName}){
            $maxWordFrequencyOfDocument =
                $hashWordDocumentCount{$bigramToken}{$documentName};
        }
    }else{
        $hashWordDocumentCount{$bigramToken}{$documentName} = 1;
        if($maxWordFrequencyOfDocument < 1){
            $maxWordFrequencyOfDocument = 1;
        }
    }
    $previousWord = $word;
}
```

Figure 6: Code Snippet for Bigram Selection and Frequency Count

### 5.3 Results

We performed experiments using unigram, bigram, trigram and quadragram feature sets. The unigram and quadragram results produce accuracy levels below 25% and are subsequently dropped. Here we present the results of bigram and trigram feature sets. The corpus contains articles on Bill Clinton, Tony Blair, and Ehud Barak.

#### 5.3.1 Label generation using bigram as the feature set

Table 6 shows the labels generated by the bigram feature set with log-likelihood as the degree of association between the features.

We now apply the Label Evaluation technique described earlier. The result indicates that the topics for Cluster0, Cluster1, and Cluster2 are Ehud Barak, Bill Clinton and Tony Blair, respectively. The overall accuracy for these labels is 42.01%.

```

foreach my $word (@wordArray) {
    $word = lc($word);
    if($previousPreviousWord eq ""){
        $previousPreviousWord = $word;
        next;
    }
    if($previousWord eq ""){
        $previousWord = $word;
        next;
    }
    $trigramToken = $previousPreviousWord." ".$previousWord." ".$word;
    if($shashWordDoucmentCount{$trigramToken}{$documentName}){
        $shashWordDoucmentCount{$trigramToken}{$documentName} =
            $shashWordDoucmentCount{$trigramToken}{$documentName} + 1;
        if($smaxWordFrequencyOFDocument <
            $shashWordDoucmentCount{$trigramToken}{$documentName}){
            $smaxWordFrequencyOFDocument =
                $shashWordDoucmentCount{$trigramToken}{$documentName};
        }
    }else{
        $shashWordDoucmentCount{$trigramToken}{$documentName} = 1;
        if($smaxWordFrequencyOFDocument < 1){
            $smaxWordFrequencyOFDocument = 1;
        }
    }
    $previousPreviousWord = $previousWord;
    $previousWord = $word;
}

```

Figure 7: Code Snippet for Trigram Selection and Frequency Count

Table 6: Labels Generated Using Bigram Features

Labels Type	Cluster 0	Cluster 1	Cluster 2
<b>Descriptive</b>	Yasser Arafat, White House, Prime <head>B_T_E</head>, York Times, Middle East, Minister Minister, Camp David, Minister <head>B_T_E</head>, United States, Prime Minister	Yasser Arafat, George Bush, Al Gore, White House, Middle East, York Times, Camp David, President <head>B_T_E</head>, United States, former President	George Bush, Bill Clinton, Al Gore, prime minister, White House, Ronald Reagan, York Times, Ariel Sharon, Camp David, United States
<b>Discriminati ng</b>	Prime <head>B_T_E</head>, Minister Minister, Minister <head>B_T_E</head>, Prime Minister	President <head>B_T_E</head>, former President	Bill Clinton, prime minister, Ronald Reagan, Ariel Sharon

Tables 7 and 8 present the corresponding contingency matrix and the application of the Hungarian algorithm, respectively.

Table 7: Similarity Scores Generated for Each Label Against Each Topic

Original Contingency Matrix			
	Bill Clinton	Ehud Barak	Tony Blair
Cluster0	20	92	113
Cluster1	50	4	51
Cluster2	112	92	142

Table 8: Contingency Matrix Showing the Diagonal Alignment after Hungarian Algorithm

Contingency Matrix after Hungarian Algorithm			
	Ehud Barak	Bill Clinton	Tony Blair
Cluster0	92	20	113
Cluster1	4	50	51
Cluster	92	112	142

### 5.3.2 Label generation using trigram as the feature set

This section presents an analogous view for the trigram case. Table 9 shows the labels generated by the trigram feature set with log-likelihood as the degree of association between the features. Table 10 shows the original contingency matrix, and Table 11 shows the application of the Hungarian algorithm to it.

The trigram experiment gives the same result as that of the bigram experiments in terms of topic selection. The topics for Cluster0, Cluster1, and Cluster2 are Ehud Barak, Bill Clinton and Tony Blair, respectively. The overall accuracy for trigram labels is 48.05%. The accuracy of the results indicates that there is an improvement of 6.04

### 5.3.3 Label generation using n-grams as the feature set with $n \geq 4$

With  $n$  greater than or equal to 4, most of the  $n$ -gram features occur only once, which introduces lots of noise in the dataset. Applying a measure of association or any other technique at this point is ineffective.

Table 9: Labels Generated Using the Trigram Features

Labels Type	Cluster 0	Cluster 1	Cluster 2
<b>Descriptive</b>	_ Minister <head>B_T_E</head>, The Minister <head>B_T_E</head>, Jerusalem Minister <head>B_T_E</head>, Israeli Minister <head>B_T_E</head>, London Minister <head>B_T_E</head>, Prime Minister <head>B_T_E</head>, Prime Minister Britain, President Minister <head>B_T_E</head>, British Minister <head>B_T_E</head>, In Minister <head>B_T_E</head>	August President <head>B_T_E</head>, President President <head>B_T_E</head>, Former President <head>B_T_E</head>, S President <head>B_T_E</head>, When President <head>B_T_E</head>, 1999 President <head>B_T_E</head>, _ President <head>B_T_E</head>, U President <head>B_T_E</head>, Clinton President <head>B_T_E</head>, former President <head>B_T_E</head>	New York TEL, The New York, New York NYT1, New York NYT18, New York NYT17, New York Times, New York Features, George W Bush, New York NEW, New York Special
<b>Discriminating</b>	_ Minister <head>B_T_E</head>, The Minister <head>B_T_E</head>, Jerusalem Minister <head>B_T_E</head>, Israeli Minister <head>B_T_E</head>, London Minister <head>B_T_E</head>, Prime Minister <head>B_T_E</head>, Prime Minister Britain, President Minister <head>B_T_E</head>, British Minister <head>B_T_E</head>, In Minister <head>B_T_E</head>	August President <head>B_T_E</head>, President President <head>B_T_E</head>, Former President <head>B_T_E</head>, S President <head>B_T_E</head>, When President <head>B_T_E</head>, 1999 President <head>B_T_E</head>, _ President <head>B_T_E</head>, U President <head>B_T_E</head>, Clinton President <head>B_T_E</head>, former President <head>B_T_E</head>	New York TEL, The New York, New York NYT1, New York NYT18, New York NYT17, New York Times, New York Features, George W Bush, New York NEW, New York Special

Table 10: Similarity Scores Using Trigrams

Original Contingency Matrix			
	Bill Clinton	Ehud Barak	Tony Blair
Cluster0	11	209	242
Cluster1	132	0	55
Cluster2	132	0	66

Table 11: Hungarian Alignment Using Trigrams

Contingency Matrix after Hungarian Algorithm			
	Ehud Barak	Bill Clinton	Tony Blair
Cluster0	209	11	242
Cluster1	0	132	55
Cluster	0	132	66

## 5.4 Summary

The conclusion we draw from these n-gram experiments is that the bigram and trigram features selection give a more reliable indication of the topic of a cluster than unigrams. Values of  $n > 4$  appear unwieldy and unusable in the main. Using  $n = 4$  appears to be of little value in the context of these experiments.

## 6 Using Tf-idf for Cluster labeling

### 6.1 Description

Term frequency and inverse document frequency are widely used basic concepts in information retrieval [22]. Term frequency is a simple technique that counts the occurrence of each term in a document. It can reveal the terms that “dominate” the document and give insight into its content. If we list the top  $n$  (perhaps 10 or 25) most frequent terms in a document, there is a good chance that a reader can recognize its topic. If not, he may still have an idea of the topic or at least recognize the domain of the document.

An obvious concern with respect to term frequency is that the most frequent terms in English text are very general in nature and tell us nothing about the content of the document (such as *the, a, an, in, of,* etc.). These words are non-substantive and may be found in the stop list. The remaining high frequency terms are of interest in determining document content. But the importance of such a word is mitigated by its distribution throughout the collection. In order to determine the importance of a term with respect to a document, we are interested in both its frequency within the document (its term frequency) and in its frequency throughout the collection (its document frequency or the number of documents in which it occurs). Tf-idf weighting considers that the most valuable terms are those with high frequency within a document and low frequency across the collection as a whole. Words that occur in large numbers of the documents can't be used to uniquely identify a document. So tf-idf weights a term based on a function of its term frequency and its inverse document frequency. A logarithmic function applied to idf values may be used to improve their decimal values.

Given the term frequency and inverse document frequency of a word, tf-idf term weighting multiplies the two values to get results that give us better sense of the importance of the term in the document. (The top-ranked words after applying tf-idf weighting [25] will not contain stop words as unigrams, since term frequency pushes the most frequent words in the document up in the list and inverse document frequency pushes those words which occur in most of the documents down the list.) The formulas used to calculate tf, idf, and tf-idf are shown in Fig. 1.



Table 12: The Calculation of Term Frequency, Inverse Document Frequency and TF-IDF [22]

$TF(t, d, D) = 0.5 + \left( \frac{(0.5 * f(t, d))}{\max(f(w, d): w \rightarrow D)} \right)$	---	(1)
$IDF(t, D) = \log \left( \frac{ D }{(1 +  d \rightarrow D : t \rightarrow d )} \right)$	---	(2)
$TFIDF(t, D) = TF(t, d, D) \times IDF(t, D)$	----	(3)
<i>Where, <math>f(t, d)</math></i>	=	<i>frequency of term within a document</i>
<i><math>\max(f(w, d))</math></i>	=	<i>maximum frequency of a word in a document.</i>
<i><math> D </math></i>	=	<i>total number of documents</i>
<i><math> d \rightarrow D : t \rightarrow d </math></i>	=	<i>number of documents containing the term t</i>
<i><math>TF(t, d, D)</math></i>	=	<i>term frequency for term t in document d and in corpus D</i>
<i><math>IDF(t, D)</math></i>	=	<i>inverse document frequency for term t in corpus D</i>
<i><math>TF-IDF(t, D)</math></i>	=	<i>term frequency-inverse document frequency for term t in corpus</i>

## 6.2 Experimental Setup

We now have a collection in which the terms are tf-idf weighted (and stop words, essentially noise in the cluster-labeling process, are no longer of concern).

Each cluster is considered as the corpus D. Each paragraph or article instance is considered a single document. The term is chosen based on the value n of Ngram. Here we show results for the bigram and trigram feature sets, so terms will be two and three consecutive words, respectively. We first calculate the frequency of each term in each document using equation 1. We also calculate the inverse document frequency of these terms using equation 2. The TF-IDF value of each term for a given document is calculated by multiplying the term frequency and inverse document frequency for that term. Finally, we aggregate these TF-IDF values of each term by summing their TF-IDF value against each document in which it occurs.



## 6.2.1 Algorithm used in Tf-idf Process

Below is the algorithm that we employ for cluster labeling using the tf-idf approach. Each file contains one cluster (the file name is passed as the command line argument). Each cluster is composed of multiple documents as indicated by the instance tag in Figure X, below. The entry associated with document id '00021' shows the text of the document between the context tags.

```
<cluster id="0">
  <instance id="00021">
    <context>
      U S President <head>B_T</head> on Tuesday nominated Alan Greenspan
      to serve a fourth term as chairman of the Federal Reserve Board
      saying that his leadership had inspired confidence not only here in
    </context>
  </instance>
  .... More instance tags goes here ....
</cluster>
```

Figure 8: A Typical Document Entry in a Cluster File

Given the cluster file whose name is passed as the command line argument, our tf-idf cluster-labeling algorithm is as follows.

For each instance in the file:

1. Locate the text of the current document within the context tags
2. Identify the n-gram tokens that make up the text
3. Store each token of the document in the hash as:

```
if ( hashWordDocumentCount {word} {document} ) {
    hashWordDocumentCount {word} {document} =
    hashWordDocumentCount {word} {document} + 1;
} else {
    hashWordDocumentCount {word} {document} = 1;
}
```

4. Store the max word count of each document as:

$$\text{hashMaxWordCountInDoc}\{\text{document}\} = \text{maxWordCountOfThatDoc};$$

5. The number of documents in the corpus = total number of (<context> tag)

6. The term frequency for a term is calculated using the hash as:

$$\text{TF}(t, d, D) = 0.5 + (0.5 * f(t, d) / \max(f(w, d) : w \rightarrow D))$$

7. Inverse Document Frequency for a term is calculated as:

$$\text{IDF}(t, D) = \log(|D| / (1 + |d \rightarrow D : t \rightarrow d|))$$

8. Finally, TF-IDF for a term is calculated as:

$$\text{TF-IDF}(t, D) = \text{summation for all document}(\text{TF}(t, d, D)) * \text{IDF}(t, D)$$

The top 20 terms calculated using step 8 become the label for the cluster.

Once the same operations have been performed on the remaining files of clusters to create the corresponding descriptive labels for those clusters, we compare the descriptive labels of each cluster to generate the unique terms for that cluster. These terms become the discriminating labels.

### **6.3 Code Snippets**

In this section, we present the code snippets used for calculation of term frequency (TF), inverse document frequency (IDF) and finally TF-IDF.

### **6.4 Results: Bigram and Trigrams with Tf-idf**

As explained previously, the selection of unigram, bigram and trigram features influence the results of the label selection process. Here, we have used the bigram and trigram features to perform TF-IDF experiments.

```

#####
# The function to do the calculation of term frequency.
# @argument1 : It will use the two global hashes for the calculation.
# @return : Nothing.
# @description :
# The term frequency for a word is calculated as below:
#  $TF(t,d) = 0.5 + (0.5 * f(t,d) / \max(f(w,d) : w \rightarrow D)$ 
#  $f(t,d)$  = frequency of term with a document
#  $\max(f(w,d))$  = maximum frequency of a word in a document.
#  $f(t,d)$  : This will be present in the hash %hashWordDoucmentCount
#  $\max(f(w,d))$  : This will be present in the hash %hashMaxWordCountInDoc
#  $TF(t,d)$  : This will be stored in the hash %hashTermFrequencyInDocument
#####
sub calculateTermFrequency{
  foreach my $outerKey (sort keys %hashWordDoucmentCount){
    foreach my $innerKey (sort keys %{$hashWordDoucmentCount{$outerKey}}){
      if($hashMaxWordCountInDoc{$innerKey} != 0){
        $hashTermFrequencyInDocument{$outerKey}{$innerKey}
          = 0.5 + ( 0.5 * $hashWordDoucmentCount{$outerKey}{$innerKey}
                    / $hashMaxWordCountInDoc{$innerKey} );
      }else{
        $hashTermFrequencyInDocument{$outerKey}{$innerKey} = 0.5;
      }
    }
  }
}

```

Figure 9: Code to Calculate Term Frequency

```

#####
# The function to do the calculation of inverse document frequency.
# @argument1 : It will use the one global hash and another global variable
# for the calculation.
# @return : Nothing.
# @description : The inverse document frequency for a word is calculated as below:
#  $IDF(t) = \log(|D| / (1 + |d \rightarrow D : t \rightarrow d|))$ 
#  $|D|$  = total number of a document
#  $|d \rightarrow D : t \rightarrow d|$  = number of document containing the term t
#  $|D|$  :: This will be present in the global variable, %noOfDocumentInCorpus
#  $|d \rightarrow D : t \rightarrow d|$  :: This will be present in the global hash, %hashWordDoucmentCount
#  $IDF(t)$  :: This will be stored in the hash %hashInverseDocFrequency
#####
sub calculateInverseDocumentFrequency{
  foreach my $outerKey (sort keys %hashWordDoucmentCount){
    my $documentCountContainingTerm = scalar keys %{$hashWordDoucmentCount{$outerKey}};
    $hashInverseDocFrequency{$outerKey} =
      log( $noOfDocumentInCorpus / (1 + $documentCountContainingTerm) );
  }
}

```

Figure 10: Code to Calculate Inverse Document Frequent

```

#####
# The function to do the calculation of TFIDF for a term
# @argument1 : It will use the two global hashes to do the calculation
# @return : Nothing.
# @description : The TFIDF for a word is calculated as below:
# TFIDF(t,d) = TF(t,d) * IDF(t)
# TF(t,d) :: Term Frequency for a term in a document, stored
# in %hashTermFrequencyInDocument
# IDF(t) :: Inverse document frequency of a term in corpus,
# stored in %hashInverseDocFrequency
# TFIDF(t,d) :: This will be stored in the hash %hashTFIDF
#####
sub calculateTFIDF{
  foreach my $outerKey (sort keys %hashWordDocCount){
    foreach my $innerKey ( sort keys %{$hashWordDocCount{$outerKey}}){
      my $score = $hashWordDocCount{$outerKey}{$innerKey} *
        $hashInverseDocFrequency{$outerKey};
      $hashTFIDF{$outerKey}{$innerKey} = $score;
      if($hashTFIDFNoDoc{$outerKey}){
        if($hashTFIDFNoDoc{$outerKey} < $score){
          $hashTFIDFNoDoc{$outerKey} = $score;
        }
      }else{
        $hashTFIDFNoDoc{$outerKey} = $score;
      }
    }
  }
}

```

Figure 11: Code to Calculate TF-IDF

Below are the labels originally generated by SenseClusters using log-likelihood and the TF-IDF technique with bigram and trigram feature sets. The corpus consists of articles on Bill Clinton, Tony Blair, and Ehud Barak.

Table 13: Labels Generated by SenseClusters using Bigrams

Labels Type	Cluster 0	Cluster 1	Cluster 2
<b>Descriptive</b>	Yasser Arafat, White House, Prime <head>B_T_E</head>, York Times, Middle East, Minister Minister, Camp David, Minister <head>B_T_E</head>, United States, Prime Minister	Yasser Arafat, George Bush, Al Gore, White House, prime minister, York Times, Ariel Sharon, Camp David, President <head>B_T_E</head>, United States	George Bush, Yasser Arafat, White House, prime minister, West Bank, York Times, Camp David, Prime Minister <head>B_T_E</head>, President Bush, President Clinton
<b>Discriminating</b>	Prime <head>B_T_E</head>, Middle East, Minister Minister, Minister <head>B_T_E</head>, Prime Minister	Al Gore, Ariel Sharon, President <head>B_T_E</head>	West Bank, Prime Minister <head>B_T_E</head>, President Bush, President Clinton

We can see that the quality of labels improves when we are using TF-IDF to generate the top

Table 14: Labels Generated by TF-IDF using Bigrams

Labels Type	Cluster 0	Cluster 1	Cluster 2
<b>Descriptive</b>	new york, prime minister, jerusalem, white house, israeli prime, camp david, george bush, former president, yasser arafat, prime minister, president bush, president clinton, column washington, al gore, british prime, ariel sharon, israel politics, peace process, middle east, palestinian leader, at camp, peace talks, and palestinian, leader yasser,	prime minister, u s, former president, al gore, white house, prime minister, in 1992, george w, british prime, israeli prime, the president, the israeli, of britain, united states, eight years, bill clinton, vice president, the democrats, the world, the former, in washington, years ago, the american, the west, of israel	prime minister, white house, new york, u s, israeli prime, yasser arafat, camp david, former president, palestinian leader, al gore, president bush, george w, the palestinian, w bush, leader yasser, president clinton, bill clinton, the united, and palestinian, british prime, middle east, of israel, ariel sharon, united states, the palestinians
<b>Discriminating</b>	jerusalem, george bush, column washington, israel politics, at camp, the israeli, peace process, peace talks	in 1992, the president, the democrats, the world, the former, in washington, years ago, the american, the israeli, of britain, eight years, vice president	the palestinian, the palestinians, the united, w bush

bigram and trigram features as labels. We also observe that the quality of labels improves when using trigrams over bigrams. These observations reinforce our previous views on n-gram feature selection.

We have run our label evaluation program on the original SenseClusters labels and on the labels generated by the tf-idf technique using bigram and trigram features. We have provided gold standard topics and asked the system to get the data on these topics from Wikipedia. Further, we have asked the system to generate its best mapping between the clusters and the labels.

Table 16 contains the summary of the result of all three cases. Beneath the final contingency matrix of similarity scores for tf-idf is the mapping generated by the system. These results are shown for tf-idf with trigrams, tf-idf with bigrams, and SenseCluster results.

The tf-idf results with bigrams and trigrams as features match each other with small differences in accuracy. However, these results do not match with result of labels generated by SenseClusters. To verify this pattern, we performed another set of experiments with a different dataset. The dataset [12] contains articles on the gold standard topics California, India, Mexico and Peru. We use SenseClusters to cluster this dataset. The clustered data is then labeled with SenseClusters using



Table 15: Labels Generated by TF-IDF using Trigrams

Labels Type	Cluster 0	Cluster 1	Cluster 2
<b>Descriptive</b>	new york times, israeli prime minister, george w bush, the white house, column washington, british prime minister, u s president, times news service, times special features, york times special, leader yasser arafat, jerusalem prime, politics jerusalem, thompson column washington, israel politics jerusalem, washington news, israeli prime minister, newspaper ltd distributed, jacoby column undated, the peace process, palestinian leader yasser, the middle east	george w bush, british prime minister, the white house, israeli prime minister, the united states, u s president, the west bank, the middle east, british prime minister, the israeli prime, the bush administration, britain's prime minister, eight years ago, american soul what, and unmistakable way, choir actually choir, colin powell could, facing the risk, great and unmistakable, heavy artillery of, neither george w, risk of stumbling, the american soul, the heavy artillery, the clinton administration	prime minister <head>b_t_e</head>, israeli prime minister, the white house, the new york, former president <head>b_t_e</head>, british prime minister, new york times, leader yasser arafat, at camp david, george w bush, <head>b_t_e</head> of israel, the united states, the palestinian leader, palestinian leader yasser, <head>b_t_e</head> of britain, the middle east, the white house, u s president, the peace process, and athlete ford, august 1999 president, his wife betty, house in august, in presenting ford, medal of freedom
<b>Discriminating</b>	by jeff jacoby, charles m sennott, column washington, israel politics jerusalem, jacoby column undated, jerusalem prime, newspaper ltd distributed, politics jerusalem, thompson column washington, times news service, times special features, washington news, york times special	american soul what, and unmistakable way, britain's prime minister, choir actually choir, colin powell could, eight years ago, facing the risk, great and unmistakable, heavy artillery of, neither george w, risk of stumbling, the american soul, the bush administration, the clinton administration, the heavy artillery, the israeli prime, the west bank	<head>b_t_e</head> of britain, <head>b_t_e</head> of israel, and athlete ford, at camp david, august 1999 president, former president <head>b_t_e</head>, his wife betty, house in august, in presenting ford, medal of freedom, prime minister <head>b_t_e</head>, the new york, the palestinian leader

trigram features, SenseClusters using bigram features, tf-idf using trigram features and tf-idf with bigram features. The results are shown in Tables 17 and 18. Tf-idf with bigram features produces the same mapping as that of tf-idf with trigram features. Also, their accuracies are comparable. In accordance to previous trends, SenseClusters results do not match tf-idf results. Also, all the accuracies are similar in all the cases with little improvement in tf-idf labels over SenseClusters

Table 16: Similarity Scores of Labels Assigned using TF-IDF-Trigram, TF-IDF-Bigram and Original SenseClusters Labeling and Final Conclusion of Label Assignment of Various Options

	TF-IDF With Trigram			TF-IDF With Bigram			SenseClusters Labels		
	Bill Clinton	Tony Blair	Ehud Barak	Ehud Barak	Tony Blair	Bill Clinton	Ehud Barak	Bill Clinton	Tony Blair
Cluster0	130	148	95	89	164	89	31	86	59
Cluster1	83	206	103	26	101	86	9	79	44
Cluster2	199	211	154	31	60	65	13	24	28
Results	Cluster0 - Bill Clinton Cluster1 - Tony Blair Cluster2 - Ehud Barak			Cluster0 - Bill Clinton Cluster1 - Tony Blair Cluster2 - Ehud Barak			Cluster0 - Ehud Barak Cluster1 - Bill Clinton Cluster2 - Tony Blair		
Accuracy	36.87%			35.86%			33.00%		

labels.

Table 17: Similarity Scores of Labels Assigned using SenseClusters-Bigram, SenseClusters-Trigram Labeling

	SenseClusters With Trigram				SenseClusters With Bigram			
	Mexico	India	Peru	California	Mexico	India	Peru	California
Cluster0	2963	1840	2147	2026	203	34	137	90
Cluster1	348	418	295	535	726	503	391	521
Cluster2	139	143	164	189	459	266	420	432
Cluster3	207	184	214	293	756	534	561	744
Results	Cluster0 - Mexico Cluster1 - India Cluster2 - Peru Cluster3 - California				Cluster0 - Mexico Cluster1 - India Cluster2 - Peru Cluster3 - California			
Accuracy	26.79%				23.97%			

Table 18: Similarity Scores of Labels Assigned using TFIDF-Trigram, TFIDF-Bigram Labeling

	TFIDF With Trigram				TFIDF With Bigram			
	Mexico	Peru	India	California	Mexico	Peru	India	California
Cluster0	341	77	242	44	154	88	44	110
Cluster1	1078	1173	509	769	442	209	374	502
Cluster2	741	626	662	432	342	176	352	450
Cluster3	921	683	662	795	319	143	376	517
Results	Cluster0 - Mexico Cluster1 - Peru Cluster2 - India Cluster3-California				Cluster0 - Mexico Cluster1 - Peru Cluster2 - India Cluster3-California			
Accuracy	30.46%				27.59%			

## 6.5 Summary

We believe that tf-idf helps us to get more relevant terms in labels. We conclude that the labels generated by tf-idf using trigrams outperform the labels generated by tf-idf with bigrams and SenseClusters [21].



## 7 Conclusion

We have worked on the fascinating area cluster labeling and label evaluation. We have explored existing mechanisms in both areas in an effort to produce our own technique. Our label evaluation mechanism gives the user a flexible, extensible and reliable option to measure the correctness of labels. By providing trustworthy feedback, our label evaluation system has played a vital role in improving the label mechanism itself. To improve the labeling technique we have examined various approaches such as using different values of  $n$  in the Ngram feature list and incorporating tf-idf to get the better terms for labels. While both unigrams and bigrams were used earlier for feature lists, our experiment shows that trigram features have improved the results and are worth considering when making the choice for feature selection. Further, tf-idf appears promising with respect to other methods of association like log-likelihood, T-Score, etc. We have also tried to improve the human readability of the labels by using the external resources such as Wikipedia.

## 8 Future Work

Finding the correct, concise and human readable labels for a cluster is an exciting area of research. In this thesis, we have discussed several possible approaches for improving the labeling technique. There is clearly a large window for improvement here. A possible method of improvement is described below:

1. Indexing the Wikipedia data [4]. This needs a high performance crawler with active crawling for new entries. Then we need an effective mechanism to manage the created indexes [9].
2. Expanding the terms. For each term in the label, where a term is based on the value of  $n$  in Ngram) get its details from Wikipedia.
  - (a) If the term is one of the topics of the Wikipedia page then simply use data of that page.
  - (b) Otherwise, use the page that is best fit for that term as per the indexing.
3. Analyze the result of the previous step using a graph algorithm. The idea here is to find the topic where all the terms of a label are converging.

See, for example, Figure 12, below.

*If we get a label for a cluster such as “Iraq War, US President, weapon of mass destruction, white house, 1993 to 2001 ”, then all these terms are pointing toward the topic “Bill Clinton”.*

4. Repeat steps 1, 2, and 3 with another external data source such as Wordnet [19].
5. Compare these two results to confirm the conclusion. Use of two external systems helps in getting independent results. If the results match, the conclusion will be more reliable and trustworthy.
6. Consider also applying different weights to the results obtained from different external systems. Different systems have different levels of data about various topics. Some sources may have extensive information on a large number of topics; others may offer only limited data.

There is clearly much room for improvement in cluster labeling and label evaluation. Our results indicate that using trigrams in this process may well be a useful avenue of expansion.

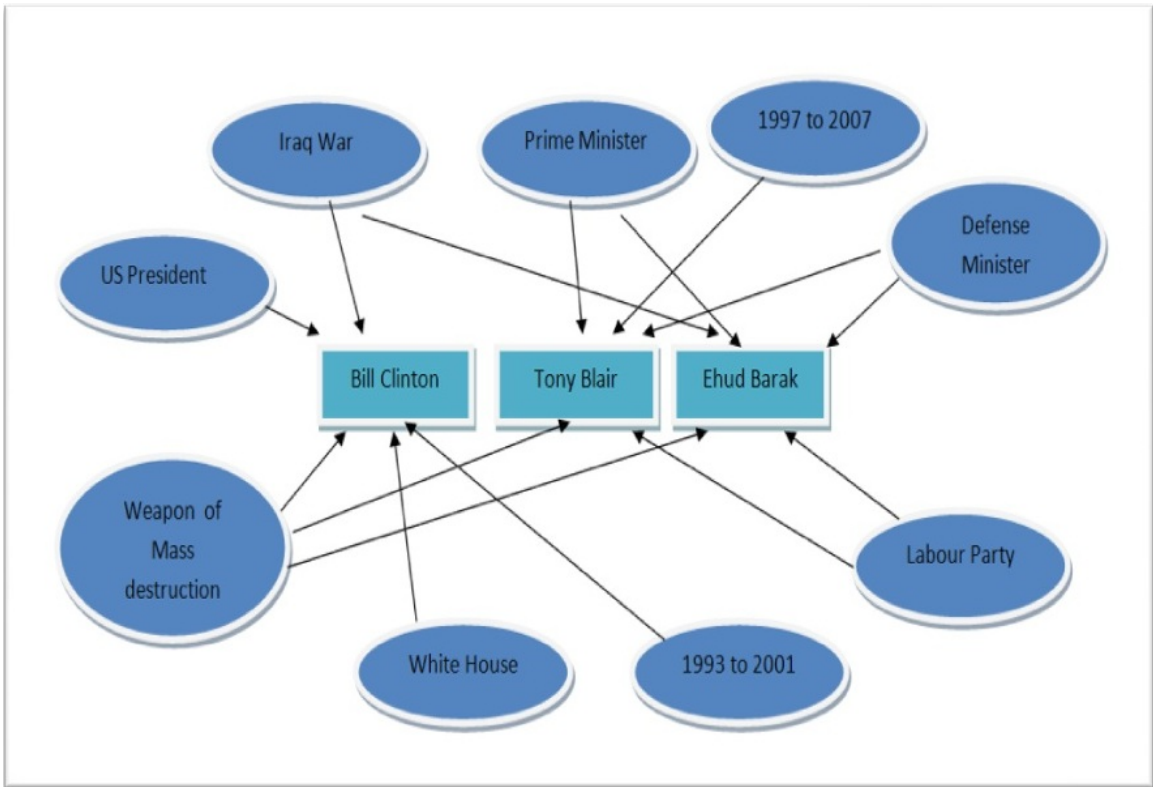


Figure 12: Visualization of 'Finding the Topics Using Terms of the Labels'

## References

- [1] CPAN. <http://www.cpan.org>.
- [2] Satanjeev Banerjee and Ted Pedersen. Extended Gloss Overlaps as a Measure of Semantic Relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence IJCAI-2003*, pages 805–810, 2003.
- [3] Don Blaheta and Mark Johnson. Unsupervised learning of multi-word verbs. In *Proceedings of the 39th Annual Meeting of the ACL*, pages 54–60, 2001.
- [4] David Carmel, Haggai Roitman, and Naama Zwerdling. Enhancing Cluster Labeling Using Wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 139–146, 2009.
- [5] Rich Caruana and Alexandru Niculescu-Mizil. An Empirical Comparison of Supervised Learning Algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.
- [6] Kenneth Church and William Gale. A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics*, 19(1):75–102, 1993.
- [7] Kenneth Church, William Gale, Patrick Hanks, and Donald Hindle. Using Statistics in Lexical Analysis. In *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, pages 115–164, 1991.
- [8] Kenneth Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.
- [9] Doug Cutting and Jan Pedersen. Optimization for dynamic inverted index maintenance. In *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 405–411, 1989.
- [10] Ted Dunning. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

- [11] Hung Hoang, Su Kim, and Min Kan. A Re-examination of Lexical Association Measures. In *Proceedings of the 2009 Workshop on Multiword Expressions*, pages 31–39, 2009.
- [12] Anagha Kulkarni. Unsupervised Discrimination and Labeling of Ambiguous Names. In *Proceedings of the ACL Student Research Workshop (ACLstudent-05)*, pages 145–150, 2005.
- [13] Anagha Kulkarni and Ted Pedersen. Name Discrimination and email clustering using Unsupervised clustering and Labeling of Similar Contexts. In *2nd Indian International Conference on Artificial Intelligence (IICAI-05)*, pages 703–722, 2005.
- [14] Anagha Kulkarni and Ted Pedersen. SenseClusters: Unsupervised Clustering and Labeling of Similar Contexts. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 105–108, 2005.
- [15] Robert Moore. On Log-Likelihood-Ratios and the Significance of Rare Events. In *Proceedings of EMNLP 2004*, pages 333–340, 2004.
- [16] James Munkres. Algorithms for the assignment and transportation problems. *Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.
- [17] Michael Oakes, Robert Gaaizauskas, and Helene Fowkes. A Method Based on the Chi-Square Test for Document Classification. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 440–441, 2001.
- [18] Ted Pedersen. Fishing For Exactness. In *Proceedings of the South Central SAS User’s Group (SCSUG-96) Conference*, pages 188–200, 1996.
- [19] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. WordNet::Similarity - Measuring the Relatedness of Concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pages 1024–1025, 2004.
- [20] Alexandrin Popescul and Lyle Ungar. Automatic labeling of document clustering. Unpublished manuscript, 2000.
- [21] Amruta Purandare and Ted Pedersen. SenseClusters: Finding clusters that represent word senses. In *Demonstration Papers at HLT-NAACL 2004*, pages 26–29, 2004.

- [22] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
- [23] Frank Smadja, Kathleen McKeown, and Vasileios Hatzivassiloglou. Translating Collocations for Bilingual Lexicons: A Statistical Approach. *Computational Linguistics*, 22(1):1–38, 1996.
- [24] Andrija Tomovic, Predrag Janicic, and Vlado Keselj. N-gram-based Classification and Unsupervised Hierarchical Clustering of Genome Sequences. *Computer Methods and Programs in Biomedicine*, 81(2):1–35, 2006.
- [25] Ho Wu, Robert Luk, Kam Wong, and Kui Kwok. Interpreting tfidf term weights as making relevance decisions. *ACM Transactions on Information Systems*, 26(3):1–37, 2008.