A Comparative Study of Feature Vector-Based Topic Detection Schemes for Text Streams

Masafumi Hamamoto† Hiroyuki Kitagawa†,‡ Jia-Yu Pan Christos Faloutsos
†*Graduate School of Systems and Information Engineering*
‡*Center for Computational Science*
University of Tsukuba
hamamoto@kde.cs.tsukuba.ac.jp
kitagawa@cs.tsukuba.ac.jp
Computer Science Department
Carnegie Mellon University
{*jypan,christos*}*@cs.cmu.edu*

Abstract

Topic detection is an important subject when voluminous text data is sent continuously to a user. We examine a method to detect topics in text data using feature vectors. Feature vectors represent the main distribution of data and they are obtained by various data analysis methods. This paper examines three methods: Singular Value Decomposition (SVD), clustering, and Independent Component Analysis (ICA). SVD and clustering are popular existing methods. Clustering, especially, is applied to many topic detection methods. ICA was recently developed in signal processing research. In applications related to text data, however, ICA has not been compared with SVD and clustering, nor has its relationship with them been explored. This paper reports comparative experiments for these three methods and then shows properties as they apply to text data.

1. Introduction

In today's world, myriad electronic documents are interchanged via networks. And the number of services that continuously distribute text data, such as e-news, chat logs, and mail magazines are on the rise. The amount of information is increasing exponentially, and it will soon become difficult to identify what is important.

In situations where voluminous text data is continuously sent to a user, it is helpful to automatically analyze the contents of document data and to detect embedded subjects. For example, assume a user receives more than one thousand e-news or mail magazines daily. No one can read them all, but many would like to view the main subjects.

We define a specific subject as a *topic*, and a task as a *topic detection*, which discovers a set of feature words and fragments corresponding to a topic in the text data. In NIST's *Topic Detection and Tracking* (TDT) [1], which is a popular research area, definitions of "topic" and "topic detection" differ from ours. In TDT, a topic is defined as a specific event or activity plus directly related events or activities, and topic detection tasks require systems to group incoming stories into topic clusters, creating new clusters (topics) as needed [15]. However, a domain of topics is changed by the incoming text data. That means a topic indicates an event that occurred on a specific date or a general subject such as sports, movies, and so on. Our research does not take into account the difference of domains and then redefine the meaning of the topic more abstractly. Rather, we redefine the meaning of topic detection to focus on discovering feature words and fragments.

In our approach to the topic detection problem, we do not look at prior machine learning for topics or words featured in one of the topics. Nor do we consider the information from which a topic detection system can automatically recognize a segment, such as tags. In TDT research, just as with ours, prior learning is ignored. On the other hand, we use *window*s that do not depend on the existence of segments. The idea is to deal with varied formatted data uniformly. A window is a text fragment that is split from text data by some constant number of words. We set our problems to detect topics in a set of windows.

Our approach to this problem is to discover *feature vector*s from a space in which the windows are distributed. A feature vector is a vector that represents data characteristics. If the number of vocabulary in all text data is m, each window could be expressed as a m dimensional vector. If the windows have the same topic, words appearing in them are similar. Therefore, vectors corresponding to the same topic windows are distributed around one feature vector. This paper calls the feature vectors representing a topic as *topic feature vectors*. Hence our topic detection problem is treated as

Proceedings of the 2005 International Workshop on Challenges in Web Information Retrieval and Integration (WIRI'05)
0-7695-2414-1/05 $20.00 © 2005 IEEE
IEEE COMPUTER SOCIETY

a problem on how to extract topic feature vectors from a set of vectors distributed in m dimensional space.

As an idea to extract topic feature vectors from windows distribution space, we examined three data analysis methods: Singular Value Decomposition (SVD), clustering, and Independent Component Analysis (ICA). SVD is a classical data analysis method and is applied to very broad research areas. For text data, SVD is applied as Latent Semantic Indexing (LSI) [4] in information retrieval research. Clustering is also a classical method and some topic detection research is based on it [13][14]. ICA is a method that was developed for signal processing [6]. Application of ICA to topic detection has been proposed [2][7][8], but comparison with or inspection of other methods such as clustering has not been done. For this reason, many properties of topic detection methods using ICA are not clear. In this paper, we quantitatively compare these three data analysis methods within the context of topic detection, and show their properties through experimentation.

The rest of this paper is organized as follows: Section 2 reports related work and the position taken in this paper. Section 3 describes how each window is represented as a vector from the target text data. Section 4 gives the extraction methods for topic feature vectors by the each of the three data analysis methods. Section 5 presents a selection method for words or windows corresponding to a topic feature vector. Section 6 reports the experimental results and gives properties for each method. Section 7 concludes the paper.

2. Related work

A popular research project for topic detection is the NIST's Topic Detection and Tracking project cited above [1]. This is a competitive project for topic detection, topic tracking, and related work. In this project, topic detection methods using incremental clustering [14], single-linkage clustering [13] are proposed. Other topic detection research in TDT is based mainly on clustering.

In addition to research in TDT, various approaches are proposed; one is constructing a statistical model by probability of word frequency in data [9], another is formulation as a clustering problem in a class of self-organizing neural networks [11]. The proposed approach that applies ICA sets out to discover topics and characteristic words for each topic in the document set [7]. The idea is used in the research of topic detection from chat log data [2][8]. These two researches focus on the time correlation in text data, and then propose the data specific ICA algorithm.

In response to that work, this paper generalizes topic detection methods to discover topic feature vectors, and applies algorithms that are comparatively well-understood in the data analysis area. We then compare each analysis method and examine the properties of each method.

3. Processing text data

This section describes how the text data is processed and represented for analysis. Our topic detection system comprises three major tasks: (1) split text data into windows, (2) extract the topic feature vectors and (3) extract words and windows related in each topic feature vector. This section covers (1); (2) and (3) are explained in sections 4 and 5.

One of our goals is to construct a unit system that can deal with varied text data. Text data in the real world is not uniform. Take e-news data as an example; it contains news articles, but the number of articles in the data is unlimited. And chat log data is continually adding words and sentences. This type of text data is defined as *text stream* in this paper. With text stream, it is very difficult to recognize boundaries between stories. Also each data may have metadata that contains the transmitted time or category of contents.

Taking this into account, all of the target text data must be transformed into units for processing by a unit system. So we transform all text data into the simplest text stream, which is the form in which no boundaries or metadata are automatically recognized and all words are presented as a word sequence in the provided order. For this word sequence, we split the data into windows. Each window overlaps by half with the adjacent window to maintain the relationship before or after words in the word sequence.

After splitting the data into windows, we apply the Vector Space Model to each window [12]. If the vocabulary in all word sequences has m words, each word in the vocabulary is represented as $w_i (1 \leq i \leq m)$. Here the vector of the j-th window is represented as having x_{ij} at i-th dimensional value; it represents the frequency of w_i in the j-th window. x_{ij} could be represented in various ways; it is not limited to the simple count f_{ij}. We set x_{ij} to the value $log(1 + f_{ij})$. This form controls the effect when w_i appears frequently in the same window.

4. Extraction of topic feature vectors

This section explains three data analysis methods to extract topic feature vectors.

4.1. Singular Value Decomposition

Singular Value Decomposition (SVD) is a classical method for extracting feature vectors in data. Depending on the application, it is called Principal Component Analysis (PCA), Latent Semantic Indexing (LSI), or Karhunen-Loeve transformation. All are essentially the same method. With feature vectors, the distribution of data is set to maximum. But these vectors are restricted to being orthogonal between any two

IEEE COMPUTER SOCIETY

vectors. Each feature vector is known to have the property that it represents co-occurrence in the target document set [4].

4.2. Clustering

Clustering is a method to classify similar data objects into same groups or clusters. We can determine the properties of the data to analyze each cluster. There are various clustering methods, but one of the most popular algorithms is the *k-means* method [10].

In our context, the method is able to detect the topics by classifying topically same windows to apply clustering to a set of windows. We can take each vector that represents a centroid as our target feature vector. To calculate similarity between windows, we use the cosine measure, which is the inner product of the vectors divided by their norms. When the frequency fraction of each word is similar, the measure yields a large value. Conversely, if it is different, the measure yields a low value.

4.3. Independent Component Analysis

Independent Component Analysis (ICA) is a signal processing method to recover original signals from mixed signals. This method assumes that each signal is produced independently, and estimates m source signals $S = (\mathbf{s_1}, \cdots, \mathbf{s_m})^T$ and mixing coefficient matrix $A = (\mathbf{a_1}, \cdots, \mathbf{a_m})$ from m mixing signals $X = (\mathbf{x_1}, \cdots, \mathbf{x_m})^T$. $\mathbf{s_i}$, $\mathbf{a_i}$, and $\mathbf{x_i}$ denote column vectors. $\mathbf{s_i} (1 \leq i \leq m)$ is also called an independent component. These matrices have the relation $X = AS$. Estimation methods are many, but the most popular one is finding an independent component by maximizing its non-Gaussianity. Intuitively, a signal mixing many signals comes close to being a noise signal. ICA assumes that probability distribution of a noise signal gets closer to Gaussian and probability distribution of an independent signal draws away from it. For details, see [6].

A topic feature vector extraction method using ICA is shown in Figure 1. To apply ICA, we assume each window as m observed signals at a time. That is, each word in the vocabulary of input text data can be assumed as a microphone, and each window can be assumed as observation signals at a given time. In the first part of the algorithm (1.1 and 1.2), the dimension is reduced to k, which is given by a user or some estimation method. In the second part (2), we apply ICA to the dimensionally reduced data X_k and decompose it into the source signal matrix S and the mixing coefficient matrix A. The third part (3) calculates the topic feature vectors. In this part, U_k shows the map from word space to the space spanned by principal components, and A shows the topic feature vectors in the space spanned by principal components.

Input: X (word-window matrix), k (the number of topic feature vector)

1.1. Apply SVD to the input data; $X = U\Lambda V^T$.

1.2. Reduce the dimension of X to dimension k by using U_k (first k principal components).

2. Apply ICA and obtain $X_k = AS$.

3. Calculate $U_k A$ as the topic feature vectors.

Figure 1. Algorithm to calculate topic feature vectors applying ICA

5. Extracting words and windows related in topic feature vectors

After the topic feature vectors are obtained, words and windows related to each topic feature vector must be extracted. This section describes how words and windows are selected from a topic feature vector.

Each topic feature vector is represented by a vector having dimensional values that correspond to one of the words in the vocabulary of input text data. Each dimensional absolute value has the influence of the corresponding word upon the topic feature vector. Therefore, if a user requests the topic detection system to output p words from each topic feature vector, the system inspects the dimensional absolute values and returns corresponding p words in the order of decreasing dimensional absolute values.

On the other hand, to select a topic feature vector corresponding to each window, a simple method is conceivable that selects the topic feature vector having the largest cosine value. This method calculates the cosine between the window and each topic feature vector, and then returns the topic feature vector with the largest cosine value.

6. Experimentation

This section compares each feature vector extraction method using near-actual data, and shows their properties. Experimentation was made on MATLAB. SVD and k-means clustering are those included in MATLAB. ICA is from the JADE package[3].

Evaluating the appropriateness of window selection is difficult when using actual data. The difficulty arises because individual stories do not have uniform, topically characteristic words. This paper, therefore, focuses on evaluating appropriateness of the selection of topic feature vectors and words corresponding to a topic feature vector.

IEEE COMPUTER SOCIETY

Topic ID	title of the topic
TP_1	Asian Economic Crisis
TP_2	Monica Lewinsky Case
TP_3	1998 Winter Olympics
TP_4	Current Conflict with Iraq
TP_5	Superbowl '98
TP_6	National Tobacco Settlement
TP_7	India, A Nuclear Power?
TP_8	Israeli-Palestinian Talks (London)
TP_9	Anti-Suharto Violence
TP_{10}	Unabomber Theodore Kaczynski
TP_{11}	Pope visits Cuba
TP_{12}	Bombing AL Clinic
TP_{13}	Cable Car Crash
TP_{14}	Tornado in Florida
TP_{15}	Oprah Winfrey Lawsuit
TP_{16}	Sgt. Gene McKinney
TP_{17}	Viagra Approval
TP_{18}	Jonesboro shooting
TP_{19}	Rats in Space!
TP_{20}	General Motors Strike

Table 1. Topics used in the experiments

6.1. Overview of experimental data

We use data in the TDT2 corpus[5]. This corpus includes news articles produced from January to June 1998 by six data sources, including CNN Headline News and the New York Times News Services. A part of them have information with topics assigned to how well ("completely" or "briefly") the topic fits. This information is added manually. Following Experiment 1 uses the CNN articles that perfectly fit the topics listed in Table 1. Experiment 2 uses both the CNN and New York Times articles that perfectly fit the topic from TP_1 to TP_{10} in Table 1. For each of these articles, we eliminate stop words and invoke stemming. We also concatenate processed articles randomly to build the input word sequence. Random concatenation is done to eliminate the effect of time correlation in the corpus.

These topics are defined in TDT, even though we redefined the meaning of topic in Section 1. However, we assume that we can obtain the same topics as TDT in case the number of TDT's topics in the input text data is exactly estimated.

In following experimentations, the number of topics is assumed to have been estimated by some method or given by the user.

6.2. Evaluation method

We evaluate each detection method by *cosine measure*. For each topic feature vector $v_i(1 \leq i \leq k)$, we calculate the cosine between all vectors $t_1, \cdots, t_k$. $t_i(1 \leq i \leq k)$ represents the topic TP_i in Table 1. The evaluated value of a topic feature vector is the largest cosine. And we calculate the average of the evaluated value of topic feature vectors as the evaluated value of the detection method. This formulation is as follows:

$$\frac{1}{k}\sum_{i=1}^{k} cos(v_i, t_\alpha), \alpha = \underset{j}{\operatorname{argmax}}\, cos(v_i, t_j)$$

The vector representing a topic is obtained as follows: the word sequence for concatenating all articles of topic TP_i is taken to be D_i. One of the words included in the vocabulary of all input data is taken to be $w_j(1 \leq j \leq m)$. m is the number of words in the vocabulary. For each D_i, frequency of w_j is taken to be tf_{ij}. From the set of the word sequence, the number of topics of which the word sequence includes w_j is taken to be df_j. We here assign the m dimensional vector for topic TP_i as shown below. This vector yields large dimensional values when corresponding words frequently appear only in that topic.

$$t_i = (log(1+tf_{i1})log(\frac{k}{df_1}), \cdots, log(1+tf_{im})log(\frac{k}{df_m}))^T$$

For SVD and ICA, the packages yield static solutions, so we run each method once. The k-means package, on the other hand, yields dynamic solutions–solutions change for every run. We run the method using k-means clustering ten times, and obtain the average and standard deviation.

6.3. Experiment 1: articles of all topics appear uniformly

This experiment assumes the simplest case in which all text data is provided by a single source and the number of articles of the unitary topic is the same. Specifically, we randomly select 30 articles from each topic, so 600 articles of 20 topics are adopted as the input text data. We build the word sequence from this input; in this result, the number of words in the sequence becomes 31,787 and the number in the vocabulary becomes 4550. The average number of words in an original article is 53. For this word sequence, we set the window size to 16, 32, 64, 128, 192, and 256 words.

The results are shown in Figures 2. The horizontal axis in the figure is the window size, and the vertical axis is the evaluated value. We see that the extracted stemmed words when window size is the widest (256 words) in Tables 2, 3, and 4. In all tables, each column shows the extracted stemmed words from one of the topic feature vectors. The actual number of topic feature vectors is 20, but we show only five to stay within the scope of this paper. We select the vectors thought to represent topics TP_5, TP_{11}, TP_{15}, TP_{16}, and TP_{20}. Selected vectors v_1 to v_5 are sorted in the order of these topics.

IEEE COMPUTER SOCIETY

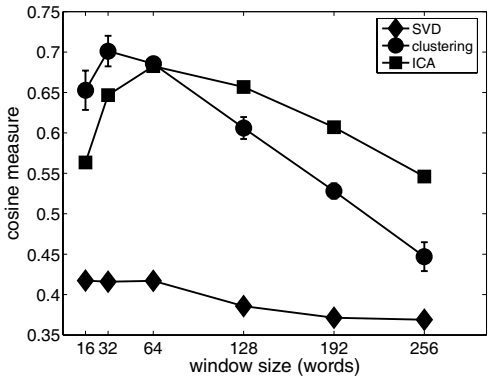

Figure 2. Evaluated value in Experiment 1

Evaluation results reveal that all methods have an appropriate window size for general trends, and evaluated values decrease when the window size is larger or smaller than the appropriate size. The reason a too wide window size is not good is that many topics are included in a window, making it difficult to find feature words in the set of windows. Conversely, a too narrow window is not good because there are too few topics in a window to find the co-occurrence relation of each topic.

Comparing the methods, at first, SVD yields worse evaluated values than ICA or clustering. Clustering yields better evaluated values than other methods for narrower windows. However, the evaluated values rapidly decrease in wider windows. ICA maintains better evaluated values than other methods, but the results in narrow windows are not as good as for clustering. We think SVD yields the worst evaluated values because the topic feature vectors given by SVD represent too general a distribution. For that reason, they cannot represent the local distribution for each topic. The results of ICA and clustering lead us to assume that ICA has a property that it considers general and more locally distributed than clustering. Hence the distribution of the window is easy to capture when window size is fairly wide and ICA yields good results, even though each window includes many topics.

Reading the extracted stemmed words, ICA yields many topically reasonable words better than the other methods. This result is achieved even when windows are wide (256 words) and each window includes many topics. Actually, SVD and clustering yield multiple topic words in a topic feature vector. Direction of other topic feature vectors is the same.

Overall, ICA is better than the other methods, except when window size is very narrow. ICA is not as sensitive to window size as clustering, so it yields good quality when window size is decided.

v_1	v_2	v_3	v_4	v_5
bowl	cuba	bomb	mckinnei	gm
super	pope	winfrei	sergeant	plant
tobacco	cuban	clinic	major	strike
test	viagra	rudolph	gene	worker
viagra	john	isra	militari	flint
columbia	castro	oprah	sexual	motor
space	paul	netanyahu	winfrei	michigan
astronaut	isra	texa	accus	viagra
asia	talk	beef	martial	winfrei
shuttl	visit	birmingham	court	team

Table 2. Extracted stems using SVD

v_1	v_2	v_3	v_4	v_5
game	pope	kaczynski	mckinnei	presid
super	kaczynski	winfrei	accus	cuba
play	cuba	judg	sexual	plant
presid	dai	trial	court	pope
bowl	peopl	dai	sergeant	dai
ve	court	oprah	major	compani
team	visit	statem	gene	cnn
plai	week	beef	former	gm
pope	havana	cnn	armi	tobacco
bronco	live	talk	top	week

Table 3. Extracted stems using clustering

6.4. Experiment 2: dealing with dual sources

The previous experiment uses text data produced by a single source. In Experiment 2, we use data produced by dual sources. The premise is that if data sources differ, the number of words in an article and its vocabulary also differ. We conducted the experiment assuming this to be the case. We used articles from CNN Headline News (CNN) and New York Times News Services (NYT) at the same time. Between these sources, article length differs significantly: CNN articles average about 51 words while NYT articles average about 411 words. For each source, we randomly select 20 articles from each topic. Thus 400 articles of 10 topics are adopted as the input text data. The number of words in the input word sequence becomes 92,204 words and the number in the vocabulary becomes 8934 words. The average number of words in an article, including both data sources, is 231 words. For this word sequence, we set window size to 32, 64, 128, 256, 384 and 512 words.

The result is shown in Figure 3. In Figure 3, we see that evaluated values decrease significantly. We think this occurs because of the increased number of words in the vocabulary. That is, evaluation calculates a cosine between two vectors in a space thousands of dimensions higher than the spaces in Experiment 1. Trends of the evaluated values are the same as in the previous experiment, although they are falling. If window size is small, then clustering yields better evaluated values than the other methods. Conversely, with large windows, ICA yields better values than the other methods.

IEEE COMPUTER SOCIETY

v_1	v_2	v_3	v_4	v_5
super	pope	winfrei	mckinnei	gm
bowl	cuba	beef	sexual	worker
bronco	cuban	oprah	sergeant	strike
game	castro	texa	accus	plant
denver	visit	talk	gene	flint
play	paul	india	major	compani
ve	john	test	armi	michigan
team	havana	nuclear	court	motor
win	ii	cattl	martial	car
plai	church	cow	misconduct	try

Table 4. Extracted stems using ICA

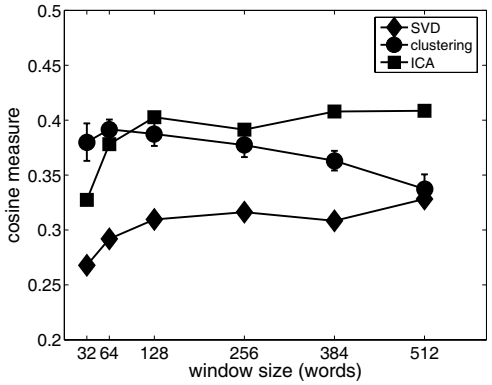

Figure 3. Evaluated value in Experiment 2

7. Conclusions and future works

This paper examined three feature extraction methods in the topic detection context: SVD, clustering, and ICA. Comparative experiments clarified the properties. SVD is the worst among the three methods. With narrow windows, clustering yields better topic feature vectors than the other methods. ICA yields better topic feature vectors with wide windows. ICA also maintains to extract more topics than the others in wide windows.

Future work will entail more detailed analysis of the properties for each method, an estimation method for the number of topics, and a method for determining appropriate window size. Our results also allow us to announce development of a method that features the good properties of clustering and ICA. The method dynamically changes from clustering for narrow windows to ICA for wide windows.

Acknowledgements

This research has been supported in part by Japan-U.S. Cooperative Science Program of JSPS, U.S.-Japan Joint Seminar (NSF grant 0318547) and the Grant-in-Aid for Scientific Research from JSPS and MEXT (#15300027, #16016205).

References

[1] TDT homepage. http://www.nist.gov/speech/tests/tdt/.

[2] E. Bingham. Topic identification in dynamical text by extracting minimum complexity time components. In *Proc. 3rd Int. Conf. Independent Component Analysys and Blind Signal Separation*, pages 546–551, San Diego, California, 2001.

[3] J.-F. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 17(1):161–164, 1996.

[4] S. Deerwester, S. Dumais, G. Furnas, and T. Landauer. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[5] J. Fiscus, G. Doddington, J. Garofolo, and A. Martin. NIST's 1998 topic detection and tracking evaluation (TDT2). In *Proc. of the DARPA Broadcast News Workshop*, pages 19–24, Hemdon, Virginia, 1999.

[6] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley, New York, 2001.

[7] A. Kabán and M. Girolami. Unsupervised topic separation and keyword identification in document collections: a projection approach. Tech. rep. 10, Dept. of Computing and Information Systems, Univ. of Paisley, 2000.

[8] T. Kolenda, L. Hansen, and J. Larsen. Signal detection using ica: Application to chat room topic spotting. In *Proc. 3rd Int. Conf. Independent Component Analysys and Blind Signal Separation*, pages 540–545, San Diego, California, 2001.

[9] H. Li and K. Yamanishi. Topic analysis using finite mixture model. *Information Processing and Management*, 39:521–541, 2003.

[10] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium in Mathematics, Statistics and Probability*, volume 1, pages 281–297, 1967.

[11] K. Rajaraman and A. Tan. Topic detection, tracking and trend analysis using self-organizing neural networks. In *Proc. 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2001)*, pages 102–107, Hong Kong, 2001.

[12] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[13] J. M. Schultz and M. Liberman. Topic detection and tracking using idf-weighted cosine coefficient. In *Proc. DARPA Broadcast News Workshop*, pages 189–192, Hemdon, Virginia, 1999.

[14] F. Walls, H. Jin, S. Sista, and R. Schwartz. Topic detection in broadcast news. In *Proc. DARPA Broadcast News Workshop*, pages 193–198, Hemdon, Virginia, 1999.

[15] C. L. Wayne. Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. In *Proc. 2nd Int. Conf. Language Resources and Evaluation*, pages 1487–1494, Athens, 2000.

IEEE COMPUTER SOCIETY