An Empirical Comparison of Four Text Mining Methods

Sangno Lee
Texas Tech University
sangno.lee@ttu.edu

Jeff Baker
American University of Sharjah
jabker@aus.edu

Jaeki Song
Texas Tech University
jaeki.song@ttu.edu

James C. Wetherbe
Texas Tech University
JCWetherbe@aol.com

Abstract

The amount of textual data that is available for researchers and businesses to analyze is increasing at a dramatic rate. This reality has led IS researchers to investigate various text mining techniques. This essay examines four text mining methods that are frequently used in order to identify their advantages and limitations. The four methods that we examine are (1) latent semantic analysis, (2) probabilistic latent semantic analysis, (3) latent Dirichlet allocation, and (4) the correlated topic model. We compare these four methods and highlight the optimal conditions under which to apply the various methods. Our paper sheds light on the theory that underlies text mining methods and provides guidance for researchers who seek to apply these methods.

1. Introduction

Modern information systems allow firms to capture vast amounts of data. Much of this data is structured data that can be analyzed using traditional database software. Increasingly, however, large amounts of data such as textual data are unstructured, and defy simple attempts to make sense of it. Manual analysis of this unstructured textual data is increasingly impractical, and as a result, text mining methods are being developed to automate the process of analyzing this data. Text mining has been used to identify intellectual core in the information systems discipline [22, 35] topic discovery [28], customer relationship management [5], and target advertising [39]. Elsewhere text mining has been used not only to identify sentiments of investors such as negative or positive opinions from business news papers and financial websites [38, 7], but also to detect objects from images [36]. Given the need to analyze and understand textual data, and given the increasing popularity of text mining methods, the time seems right for a research essay to examine the major methods available for text mining, and to address the topic of how to appropriately apply text mining techniques in IS research and practice.

This paper has two intended contributions. First, we review four methods that are commonly used in text mining research in order to open up the use of these methods to a wider audience of researchers. The four methods that we discuss are latent semantic analysis, probabilistic latent semantic analysis, latent Dirichlet allocation, and the correlated topic model. These methods range from early approaches to text mining that make use of discriminative models to more recent approaches that use generative or probabilistic models. We choose the four methods because of their wide applications and primary methods in the development of text mining, but do not consider syntactical and morphological approach of natural language because of the lagged development and relatively unsuccessful performance [25, 30, 31]. We explain that the more recently-proposed generative models overcome the limitations of previous methods and provide researchers with the ability to identify patterns in textual data through the text modeling process.

Second, we present an example that compares the use of these four types of text mining analysis. The comparison reveals the characteristics and limitations of each method, and highlights the appropriate conditions in which each method could be applied. Each of the four methods of text mining that we present here has their own unique features, and thus can be understood as optimal for a given set of tasks. Thus, our comparison provides guidance for researchers and practitioners who apply text mining methods.

The paper is organized as follows. In section two, we provide and overview of fundamental concepts and assumptions in text mining. We also explain that the vector space model can be used as a basic measure of language distance. In section three, we explain the four various methods of text mining and describe the theoretical underpinnings of each method. We also explain the advantages and disadvantages of each of the four methods. In section four, we illustrate the used of these four methods. Finally, in section five, we provide directions for future research and review our contributions.

2. Fundamental concepts in text mining

The four methods of text mining that we describe

978-0-7695-3869-3/10 $26.00 © 2010 IEEE

here are built upon a set of basic assumptions and also upon the vector space model (VSM). The VSM [32, 33, 34] provides a way to represent textual data in vector space and measure the distance of language vectors from one other.

2.1. Concepts and assumptions

The fundamental unit of text is a *word*. Words are comprised of characters, and are the basic units from which meaning is co structured. By combining a *word* with grammatical structure, a *sentence* is made. Sentences are the basic unit of action in text, containing information about the action of some subject. *Paragraphs* are the fundamental unit of composition [37] and contain a related series of ideas or actions. As the length of text increases, additional structural forms become relevant, often including sections, chapters, entire documents, and finally, a *corpus* of documents. In a text mining study, a *document* is generally used as the basic unit of analysis because a single writer commonly writes the entirety of a document and the document discusses a single topic. A document can be a paper, an essay, or a book, depending on the type of analysis being performed and depending upon the goals of the researcher. In some cases, a document may contain only a chapter, a single paragraph, or even a single sentence.

In text mining studies, the syntactical structure of a sentence or paragraph is intentionally ignored in order to efficiently handle the text. As a result, a sentence is regarded simply as a set of words, or a "*bag of words*" in the parlance of text mining. This idea that a sentence is a bag of words without additional structure is a basic assumption of most text mining. Because the syntactical structure of text is ignored in text mining, the order of words can be changed without impacting the outcome of the analysis. The bag-or-words concept is also refereed to as exchangeability in the generative language model [1, 2].

Two closely-related problems that arise in analyzing text are *synonymy*, the use of two or more different terms that mean the same thing, and *polysemy*, the fact that a single word may have multiple meanings. This problem has been addressed in a variety of ways [19]. Because different text mining techniques have slightly different approaches to dealing with synonymy and polysemy, we will return to this issue as we explain each of the various text mining techniques.

In the next section, we will look at the vector space model, another of the fundamental concepts that underlies many text mining approaches.

2.2. Vector space model

The vector space model (VSM) regards a document as a bag of words, and is mainly used for information retrieval tasks, such as keyword searches. In this model, sets of words are represented as mathematical vectors, which are one-dimensional arrays. In this model, the degree of similarity between two documents is calculated using the cosine function, a familiar mathematical function from Euclidian geometry, $\cos \theta = (v_1 v_2) / (\|v_1\| \, \|v_2\|)$. Thus, the VSM enables documents to be represented in a simple and manageable matrix form, and does so in a way that allows the distance between documents (vectors) to be measured. These two features are foundational for the four methods of text mining that we will describe in the next section.

A word that is common across all documents is of little use in distinguishing documents from one another. In contrast, a word that appears in only a small number of documents have high discriminability. Such a word is potentially very useful in discriminating between documents. To calculate the discriminability of a term over the corpus of documents, the inverse document frequency (*idf*) has been developed. The inverse document frequency for a word is calculated using $\log (N/df_i)$, where N is the number of documents, and df_i is the frequency of a word i over all documents, generally called as inverse document frequency (*idf*). This scheme is designed to consider the discriminative power of a word both in a document and over all of the documents in a corpus of documents.

The VSM has four primary limitations. First, in information retrieval, a longer-length document has a reduced similarity values to a query because the weight of document $\|D\|$ in the cosine similarity, a denominator, may be very high. As a result, a longer-length document has a reduced opportunity to math a query. Second, as we have noted earlier, the order of words in a document is still ignored and the bag of words assumption is made. The semantic structure of a document is potentially valuable information that the VSM disregards. Third, the keywords must match exactly with those in the document and the issue of synonymy is not addressed. Fourth and finally, in spite of similar meaning of documents, if two documents use different terms for the same concept, the similarity between the two documents will not be

identified. Thus, the issue of polysemy is not addressed either. To overcome the last two limitations, latent semantic analysis has been developed. It is to this method that we now turn our attention.

3. Four methods in text mining

Text mining has been developed from the discriminitave model such as latent semantic analysis to the generative model such as probabilistic latent semantic analysis, latent Dirichlet allocation, and correlated topic model. In the next section, we first examine discriminative model, and then other three generative models.

3.1. Latent semantic analysis

As we have noted, the VSM cannot deal with synonymy and polesemy with the following three reasons. First, for the synonymy, documents are only a small fraction of possible words and do not consider all word in documents [8]. Thus, words having similar meanings in a query may not match in target documents. For instance, a word, *beautiful*, may not be found in documents written using *pretty* word. Second, for the polesemy, the VSM has no any methods to consider the multiple meanings of a word [8, 21]. The VSM only consider word form and does not consider semantic meanings of a word. Third, the VSM does not take into account co-occurrence of words such as *information technology* words [8]. Because two words are frequently used in information systems documents, the consideration of co-occurrence of words provides to capture the meaning of a word. For instance, the *information* word is confined by the technology word, and *vice versa*.

To address the issues, latent semantic analysis (LSA) has been developed [8, 21, 27]. The LSA projects the original vector space or term-document matrix into small *factor* space. By exploring spareness of the original matrix, words are grouped into factors. The dimension of matrix is accomplished using singular value decomposition which decomposes the original matrix into three matrixes, a document engenvector, eigenvalues, and a term eigenvector. The original matrix can be approximated by multiplying three matrixes. Also, using the factor-loading matrix and rotation of matrix such as varimax method, words can be grouped into factors [21]. Because of orthogonal characteristic of factors, factors have little relationship with other factors, and words in a factor have a high correlation.

The factors in LSA can deal with more synonymy than polesemy. Words belonging to each factor can be regarded as synonyms because words in a factor have high correlation. However, for polysemy, the same form of a word needs to be appeared in different topics according to usage. For instance, *java* as a programming language in a factor needs to be appeared as *java* as coffee in other factors. Unfortunately, the orthogonal characteristic of factors prevent from appearing multiple appearance of a word in different topics.

Papadimitriou et al. [27] investigate optimal conditions for applying LSA, and found that LSA succeeds in capturing the underling semantics of the corpus under the following three conditions. First, there no style modifier. Second, each document is on a single topic. Third, the words can be partitioned among factors so that each topic distribution has high probability on its own words, and low probability on all others. In the information systems research, Sidorova et al. [35] apply LSA to identify intellectual cores in information systems research using abstracts of elite journals.

The LSA improves the VSM by considering synonyms but contains three limitations. First, it is a straightforward matrix dimensions and is not built upon robust probabilistic background. Second, the number of factors cannot be determined statistically, and the optimal determination of factor numbers depends on human judgments. Third, the polysemy is partially dealt with in LSA because of the orthogonal characteristic. To overcome the straightforward matrix reduction and polesemy problem, the generative methods have been developed, and in the next section, we explorer the first generative model, probabilistic latent semantic analysis.

3.2. Probabilistic latent semantic analysis

Due to the *exchangeability* of words or bag-of-words in text mining, the occurrence of words can be modeled with the probabilistic theory. Rather than depending on matrix reduction of a realized word-document matrix, the generative model sets a model firstly, and then estimate unknown parameters of the model using a realized word-document matrix. Thus, the major difference between generative models is on the document generation model and on the estimation method for unknown parameters of the models.

The probabilistic latent semantic analysis (PLSA) assumes that documents are generated throughout the following three steps [17, 18, 27]. First, a document d are generated or selected with probability $P(d)$. Second, topic z are picked with probability $P(z|d)$. Third, each word w in a topic is generated with

probability $P(w|z)$. This generative process is based on two assumptions. First, the realized word-document pairs are generated independently. Second, words w are generated on the condition of topic z and are independent to the specific document d. Thus, the probability of a word-document occurrence, $P(d,w)$ can be represented with $P(d)\ \Sigma_{z\in Z}\ P(w|z)P(z|d)$ or alternatively with $P(d,w) = \Sigma_{z\in Z}\ P(z)P(w|z)P(d|z)$ with Bayes' rule. To determine unknown parameters, $P(z)$, $P(z|d)$, and $P(w|z)$ with known information of $P(d)$, $P(w)$, and $P(d,w)$, Expectation Maximization (EM) algorithm [9, 24] is used where an expectation step (E) is to estimate posterior probabilities for latent variables z based on the current estimates of the parameters, and a maximization step (M) is to update parameters for the posterior probabilistic obtained in the E-step. The EM algorithm is a general solution to find unknown parameters in generative models. PLSA estimates a topic vector $P(z)$, a document matrix given topics $P(d|z)$, and a word matrix given topics $P(w|z)$. In PLSA, topic vector and other two document and word matrixes contain probability values, allowing probabilistic interpretation of words.

The three parts of PLSA, $P(d|z)$, $P(z)$, and $P(w|z)$ are correspondent with the three parts of LSA, document eigenvectors, eigenvalues, and words eigenvectors. However, the interpretation of values is different in that in LSA, the values of estimated parameters denote probabilities of words, but the values in LSA denote loading values for factors without meaning of probability. The topics are called in PLSA as *aspect*, but the meaning is the same with factors or topics.

Like factors in LSA, topics generated from PLSA are related with synonyms and ploysemy. In general, in the generative model, the words having the highest marginal probabilities are used in a topic at the first time, and then the words having the next marginal probabilities are used in the topic sequentially. Therefore, it is likely for generally used words such as titles words to have high probabilities in a topic. This characteristic allows generative models to capture general trend rather than unique trend in documents. Hofmann [17] and Hofmann et al. [18] apply PLSA to identify factors in broadcast news stories, bible, and abstracts of the *neural* journal and to make indexing for information retrieval. Sivic et al. [36] use PLSA for clustering images by discovering objects in images.

For synonymy of words, whereas PLSA provides topics in which words has similar meanings, LSA provides high correlations in a factor without connotation of semantic meanings. For polesemy of words, in PLSA, words can be appeared with other topics at the same time with high probabilities. However, because PLSA does not model the generation of documents, $P(d)$, it is a partial generative model. To fully implement the generative process for document level and sophisticate the generative process for topics, latent Dirichlet allocation has been proposed and that model is the topic in the next section.

3.3. Latent Dirichlet allocation

By addressing the limitations of PLSA, another generative model, latent Dirichlet allocation (LDA), can provide a probabilistic model at the level of documents [4, 13]. In LDA, each document is generated by mixing topics. Thus, for every document, we have a different distribution over the number of topics. Figuratively, this distribution is like events from a big biased dice that puts weights on one or two numbers. By throwing this dice, we can determine which topics will be used in a document. If a document is about a digital camera, the topics for the document will be mostly about hardware specification, performance, and design. In LDA, the topic mixture for documents is determined by the Dirichlet distribution [6].

According to LDA process, each document is generated the following three steps. First, the number of words used in a document is determined through sampling from the Poisson distribution [15]. Second, from the Dirichlet distribution, a specific distribution over topics is elicited. Third, based on the document-specific distribution, topics are generated, and then words for each topic are generated. For instance, assume that topic 1 has P(A) = 0.3 and P(B) = 0.7, and topic 2 has P(A) = 0.9 and P(B) = 0.1, and a document-specific distribution looks like P(T1) = 0.2 and P(T2) = 0.8. Now, we can generate 10 topics for a document like 1, 2, 2, 2, 2, 2, 2, 1, 2 and, 2. For each topic, words are elicited from each topic distribution. Then, words in a document can form like B, A, A, A, A, A, A, B and A.

Like PLSA, LDA also provides topics as well as words for each topic with probability values. Compared to PLSA, LDA has an advantage for modeling of long-length documents composed of several topics. Because LDA models a document with multiple topics, topics do not needed to be limited to single topic document in LSA and PLSA. For synonymy, words having similar meanings are clustered into a topic. Also, for polesemy, a word but different meanings can be appeared into other topics. Blei et al. [4] apply LDA to identify topics from TREC AP corpus [16] and to classify documents

using the Reuters-21578 dataset. Griffiths and Steyvers [14] use LDA to discovery topics in the Proceedings of the National Academy of Sciences (PNAS).

However, because there is no modeling for topic relationships, LDA cannot make the representation of relationships among topics. This limitation can be significant given that a word can be used in several topics simultaneously. For example, the word *java* can be used for a *coffee* topic as well as a *programming* topic. If the word *java* is used in a programming topic, it is closer to a *system* topic than a *coffee* topic. Thus, the *programming* topic has the close relationship with the *system* topic. In terms of polesemy, the topic relationships allow to identify closeness among senses of a word. For the example of *java*, the senses of *programming* and *system* of *java* can have close relationship. Therefore, by considering topic relationships, more realistic model of document can be expected, and another method of text mining addresses this limitation.

3.4. Correlated topic model

The correlated topic model (CTM) has been proposed to address the limitations of LDA [3]. CTM can explicitly address the correlations among latent topics using logistic normal distribution [23]. In LDA, it is assumed that each word in a topic comprises a multinomial distribution, and topics are shared with all documents but the proportion of topics in a document varies according to Dirichlet distribution. CTM follows the same document generative process of LDA, but the difference is in the use of logistic normal distribution rather than Dirichlet distribution.

For example, for a *programming* topic, *java*, *basic*, *pascal*, *.net*, and *c#* comprise a normal distribution with respect to the occurrence for the words. For three topics including *programming*, *system*, and *database*, the proportions of mixture comprise a Dirichlet distribution. However, under a Dirichlet distribution [6], the components of the proportions are practically independent, so topics cannot have relationships with other topics, and in turn, the distribution precludes allocating a word to several topics. In fact, if topics do not have relationships with other topics, a word in a topic also cannot have relationships with other topics. To remedy this problem, the correlated topic model employs the more flexible logistic normal distribution, considering a covariance structure among the components of proportions. Blei and Lafferty [3] show the possible applications of the CTM to identify topics in articles from the *science* journal.

For synonymy, CTM also provides topics in which words has close semantic distance. For polesemy, CTM allows a word to appear in other topics at the same time. Furthermore, due to the identification of relationships among topics, CTM can provide to capture close meanings even in the polesemy. For example, a *programming java* topic has closer relationship with a *web service* topic than a *coffee java* topic. The following table summarized the characteristics and limitations of each method in text mining.

Table 1. The characteristics and limitations of four text mining methods

Models	Characteristics / Limitations
Latent Semantic Analysis (LSA) [8, 21]	Characteristics • Reduces dimensionality of *tf-idf* using Singular Value Decomposition • Captures the semantics of words to address synonymy and polysemy • Straightforward statistical background Limitations • Difficult to determine the number of topics. • Difficult to label a topic in some cases using words in the topic
Probabilistic Latent Semantic Analysis (PLSA) [10, 12, 17, 27]	Characteristics • Models each word in a document as a sample from a mixture model • Mixture components are multinomial random variables that can be viewed as representations of "topics" • Each word is generated from a single topic; different words in a document may be generated from different topics • Each document is generated from mixing proportion of topics • Reduces dimensionality to topic level Limitations • Provide no probabilistic model at the level of documents
Latent Dirichelet Allocation (LDA) [4, 13]	Characteristics • Multinomial distribution for words in topics • Dirichelet distribution for topics Limitations • Not consider relationships between topics • Cannot allocate a word multiple topics
Correlated Topic Model (CTM) [3]	Characteristics • Considers relationships between topics using logistic normal distribution • Allows the allocation of a word to multiple topics • Allows topic graphs Limitations • Computational complexity Some problem in variational method for the inference of parameters

4. Empirical analysis: an illustrative example of topic detection.

Topic detection is a general application of almost

all text mining approaches from LSA to CTM. When the proponents of each model propose their models in their studies, their usual application of the models is in the topic detection area. Although topics are differently called with factor solutions for LSA, aspects for PLSA, and topics for LDA and CTM, they have similar characteristics in that the words in a topic have similar semantic meanings. Because all four approaches can be applied to the topic detection study directly, an experiment using the same documents can reveal similarity and difference among four models. Moreover, to our knowledge, because there has been no such experiment of the comparison, this analysis might shed light on the clarification of fortes and limitations of each approach.

For the experiment, we specifically collect users' comments on a camera (Canon PowerShot A590IS 8MP Digital Camera with 4x Optical Image Stabilized Zoom) from Amazon website because users' comments on a product can be not only valuable guidance for developing new products but also valuable materials for adjusting defects of products to companies. We collect 258 comments for the camera but exclude 4 comments because the reviews are written with Spanish language. Because the experiment requires high performance computing and capacity for estimating parameters of each method and storing vector space of word-document occurrences, we use Linux clusters and a grid computing environment in a university. In particular, for high performance, we develop a program using SVDLIB [29] for LSA, and revise programs of PLSA, LDA, and CTM that are provided in each proponent using Message Passing Interface [26].

For representing documents with vector space model, we consider word or term frequency (*tf*) and inverse document frequency (*idf*) for LSA, but consider only word frequency in generative methods because this usage is typical use in each method. The rationale behind the *tf* scheme is that the more a word is used in a document or over documents, the more the word will represent topics. On the other hand, the rationale behind the *idf* scheme is that if a word is frequently used in a specific document, but rarely used in other documents, the word can represent a distinct topic. Accordingly, if we use the *idf* scheme for topic detection, words for a topic will be composed of specific words in one document rather than generally used words over many documents. Therefore, while the *tf* scheme focuses on general topics, the *idf* scheme focuses on the distinct topics.

When drawing two vector space matrixes, we exclude stopwords such as *a*, *an*, or *the* because of little meanings of these words, and obtain a 254-3,189 dimensions of a document-by-word matrix. However, we do not apply the stemming method that considers only first few characters for similar morphological form. Because the stemming method reduces every form of a word such as noun and adjective into common character parts, the effect of occurrence of a word is mixed with different forms and the result can be hard to interpret. For instance, the word elaboration has five forms such as *elaborate*, *elaborately*, *elaboration*, *elaborative*, and *elaborator*, and if we conduct stemming for these words as "*elaborat*," all those words appeared in different position are treated as the same "*elaborat*" word, and thus the occurrence of "*elaborat*" can be over estimated. Furthermore, when we consider the major meaning of a topic, it is hard to conjecture the degree of weight of "*elaborat*" because we do not know where this word comes from primarily. We know that the adjective, *elaborately*, is less important than the noun, *elaboration*. Therefore, we do not employ the stemming method.

For the number of topics, we set it with 10 topics for four models. Although some studies suggest methods to determine the optimal number of topics [11], those are not complete solutions and still the best approach is to retain few rather than whole topics, as long as they provide a reasonable interpretation of topics [20].

Table 2 shows the 10 topics solution of LSA using *tf-idf* scheme (some topics are obligated with intention because of space limitation). To get solution, we apply normalization, singular value decomposition, factor loadings, and varimax rotation. As shown in the Table 2, words in a topic comprise distinctive or peculiar words, and topics seem not to relate each other. This result comes from three sources. First, because LSA is a dimension reduction with singular value decomposition, it does not consider semantic distance but only consider correlations in a factor and between factors. Thus, LSA cannot guarantee semantic topics solution. Second, the result can arise from *idf* scheme. Because the *idf* scheme gives more weight on distinctive words in a document rather generally used words over documents, words in factors are composed of distinctive words. Third, the result can be attributed to orthogonal characteristics of factors in LSA. Because factors are mutually exclusive, words in a topic are not appeared into other topics.

Table 2. The 10 topic solutions of LSA

T2:	T3:
play(2.3443)	fiance(2.1745)
pentax(2.3115)	proffesional(2.1671)
operates(2.2714)	recomend(2.1671)

stop(2.2649)	nephew(1.9605)
company(2.2543)	stevesdigicams(1.9605)
wish(2.2521)	disposable(1.9047)
comparing(2.2499)	fail(1.8254)
array(2.2465)	complicated(1.8141)
cards(2.2354)	love(1.7544)
mentioned(2.2133)	books(1.7350)
T7:	T8:
timer(2.2143)	agredd(2.4546)
dioramas(2.1407)	brice(2.4546)
fuzziness(2.1407)	notches(2.4546)
scale(2.1407)	sillyreviews(2.4546)
shipped(2.1147)	told(2.4165)
apparently(2.0475)	carpeted(2.3284)
street(2.0475)	desires(2.3284)
crystal(2.0453)	varying(2.3284)
lets(2.0384)	loves(2.3203)
condition(1.9827)	fortunately(2.3034)

For better understanding of the result of LSA, we draw four main document having high loading values for topic 3. Table 3 shows the contents of four documents.

Table 3. The four documents of the topic 3

Docs	Contents
#24	I gave this camera to my son and his **fiance** and they simply **love** it. It's not too **complicated**, has lots of features and takes great photos.
#70	**Love** it so much that I bought a second one for my **nephew**. The price has gone down too. Steve of ***StevesDigicams***.com loved it too. Check out the complete, very detailed review on his website.
#68	Hi. I just bought a low price Olympus camera but it was very bad, so I just bought this Cannon and I **love** it. Very easy to use, excellent photos. Very professional for a low price camera. I **recomend** it for sure. Great quality photos and very **proffesional**. regards.
#7	The Canon 590 takes fantastic photos! I've had trouble in the past with shaky pictures from other digital cameras. But with the built-in image stabilizer, they are clear and beautiful, even up close! I've gotten some great shots. I can even take close up photos of **books** to show on computer monitor and the text is clear and readable. **Love** the movie mode, too. I like the ease of point and shoot, but if you want to change manual settings are available too.

The document 24 contains *fiance*, *love*, and *complicated*, and the document 70 contains *love*, *nephew*, *stevesdigicams*. Except for *love*, other words are only appeared in each document uniquely. This feature is obvious in document 68 whose contents are composed of two wrong spelled words: *recomend* and *proffessional*. Because topics elicited from LSA prefer distinctive words with above three reasons, the wrong spelled words are loaded into topic 3 along with *love*, the connect word for document 68 with other documents. The document 7 contains *books* and *love*, and *books* is only appeared in that document. If we look at these four documents for the topic 3, we are able to realize that each document contains peculiar words that are only appeared in each document as well as few shared words among documents. For example, in the case of topic 3, the peculiar words in each document are grouped into the topic 3 because of one shared word, love. Therefore, topic detection in LSA can be used to identify distinctive features of documents rather than general words.

Papadimitriou et al. [27] investigate optimal conditions for applying LSA model and found that LSA succeeds in capturing the underlying semantics of the corpus under the following conditions; "1) there is no style modifier; 2) each document is on a single topic; and 3) the terms are partitioned among the topics so that each topic distribution has high probability on its own terms, and low probability on all others." Fortunately, in this case, each document deals with a single topic, camera, and peculiar word are allocated into a specific topic in terms of shared words, giving high probability of words for the topic. However, this optimal condition implies that if there are many topics in a document and a word cannot be partitioned among the topics, the LSA model is inappropriate. Because of the reasons, many studies applying LSA employ abstracts of papers or paragraph of documents as a unit of analysis [22, 35].

Table 4 shows the 10 topics solution of PLSA. Compared to the 10 topics solution of LSA, the solution of PLSA has three distinct characteristics. First, the title words to describe a camera's name (Canon PowerShot A590IS 8MP Digital Camera with 4x Optical Image Stabilized Zoom) such as *camera*, *cannon*, *shots*, and *amazon* are loaded for every topic. This characteristic can be illustrated with the process of PLSA. According to PLSA, the original frequency matrix can be decomposed into three unknown probability parts including document probability given a topic, topic probability, and word probability given a topic, and the three unknown parameters are estimated with known probability of document and word using EM algorithm. Thus, if a word has high probability, the use of the word in a topic can steeply increase the probability of the topic. In other words, the high marginal probability of a word can easily maximize the probability of a topic. Therefore, the words that are frequently used words with high probabilities are loaded into every topic. Second, each topic is composed of words related each other to describe the features of a camera such as battery and zoom, and an experience such as good and easy. Because the words that are used to describe the title name of the camera have the next high marginal probability, these words are loaded into each topic. For instance, *battery*, *memory*, or *sensor* is principal

hardware functions, so these words are commonly used to illustrate a camera in uses' comments. Third, the same words are loaded into different topics at the same time. For example, *batteries* are loaded in topic 1, 2, 8, and 9. This characteristic comes from polysemy feature of PLSA, the same form of a word but different meanings. Although *batteries* are used in the four different topics, its meanings are different and confined with other words in that topic. The *batteries* in the topic 0 is about the *price* of *batteries*, but the *batteries* in the topic 9 is *duration*, *type*, and *voltage* of batteries. Overall, the topics in PLSA seem to reflect a trend or theme in users' comment, but the topics in LSA reflect uniqueness in users' comment. Table 4 shows

Table 4. The 10 topics solution of PLSA

T0:	T1:
camera(0.0649)	camera(0.0747)
pictures(0.0316)	great(0.0209)
card(0.0164)	pictures(0.0166)
easy(0.0159)	digital(0.0149)
bought(0.0141)	features(0.0146)
picture(0.0135)	canon(0.0131)
good(0.0131)	cameras(0.0130)
great(0.0131)	easy(0.0128)
digital(0.0119)	point(0.0122)
batteries(0.0117)	good(0.0118)
memory(0.0101)	quality(0.0116)
amazon(0.0088)	photos(0.0111)
zoom(0.0085)	price(0.0103)
just(0.0082)	bought(0.0091)
price(0.0080)	takes(0.0083)
T8:	**T9:**
camera(0.0574)	camera(0.0621)
batteries(0.0339)	shots(0.0255)
good(0.0187)	battery(0.0232)
photos(0.0123)	flash(0.0226)
just(0.0121)	batteries(0.0212)
battery(0.0121)	nimh(0.0134)
shots(0.0118)	canon(0.0124)
took(0.0115)	time(0.0120)
great(0.0105)	indication(0.0119)
like(0.0102)	problem(0.0111)
time(0.0082)	used(0.0079)
used(0.0079)	alkalines(0.0072)
light(0.0078)	voltage(0.0063)
pictures(0.0077)	just(0.0062)
really(0.0076)	came(0.0057)

Table 5 shows the 10 topics solutions of LDA. The topic solution of LDA reflects the principal components of a camera and the users' experience. For instance, the topic 0 is about *batteries* and users' experience on batteries represented with *good* adjective. Topic 1 is about *good flash*, topic 2 is about *great picture*, topic 3 is about *good image*, topic 4 is about *easy use* of the camera, and so on. Likewise topic solution in PLSA, the principal words of a camera (camera and pictures) are highly loaded into each topic because of high marginal probabilities of the words, but words showing users' experience are closely loaded into related hardware characteristics. For instance, *good* word in topic 0 is loaded into between *batteries* and *battery* with 0.025 probability. The close appearance of adjective and noun allows researchers to labeling each topic with easy. For example, topic 0 can be named with "good battery." Thus, compared to the solution of PLSA, the topic solution of LDA permits an easy labeling, and related words are slightly closely converged into each topic.

LDA also provide better modeling of documents. For instance, the document #2 contains the following contents.

"…I would rate this camera only as average at this time because of the battery problems… Selected this camera model based on Consumer Reports' recommendation, mostly favorable user reviews on Amazon and some ther websites… The camera out of the box seemed to be functional for all the features I tested, but after playing with the camera… To reduce power use, changed some settings to turn LCD off after a short time, and the IS… A friendly camera repair place person told me about the Canon website online repair setup…"

Although the length of contents of document #2 is large, each topics are loaded into appropriate topics such battery and pictures. Therefore, LDA can provide good result despite of long-length documents and allow facile labeling of topics.

Table 5. The 10 topics solution of LDA

T0:	T1:	T2:
camera(0.0533)[1]	camera(0.0440)	camera(0.0640)
batteries(0.0244)	flash(0.0142)	great(0.0237)
good(0.0125)	good(0.0105)	pictures(0.0235)
battery(0.0114)	mode(0.0102)	price(0.0175)
canon(0.0108)	zoom(0.0095)	just(0.0132)
pictures(0.0101)	canon(0.0083)	batteries(0.0112)
picture(0.0099)	picture(0.0081)	flash(0.0112)
just(0.0099)	pictures(0.0079)	easy(0.0107)
want(0.0079)	better(0.0079)	takes(0.0096)
easy(0.0066)	manual(0.0077)	good(0.0090)
like(0.0066)	time(0.0071)	like(0.0086)
life(0.0065)	digital(0.0068)	picture(0.0085)
shots(0.0063)	great(0.0066)	love(0.0085)
great(0.0061)	batteries(0.0062)	setting(0.0082)
feature(0.0059)	shots(0.0062)	time(0.0081)
T3:	**T4:**	**T5:**
camera(0.0404)	camera(0.0721)	camera(0.0434)
good(0.0177)	great(0.0182)	photos(0.0129)
image(0.0152)	digital(0.0164)	just(0.0115)
pictures(0.0129)	easy(0.0161)	quality(0.0111)
cameras(0.0119)	canon(0.0152)	price(0.0110)
bought(0.0112)	good(0.0129)	manual(0.0100)
canon(0.0108)	batteries(0.0128)	bought(0.0097)
quality(0.0100)	card(0.0127)	great(0.0096)
noise(0.0094)	photos(0.0100)	amazon(0.0096)
just(0.0077)	price(0.0099)	product(0.0095)
easy(0.0073)	pictures(0.0097)	good(0.0091)
features(0.0071)	small(0.0096)	cameras(0.0077)

manual(0.0054)	shots(0.0093)	does(0.0069)
better(0.0050)	cameras(0.0088)	features(0.0068)
zoom(0.0049)	features(0.0081)	like(0.0067)

Finally, the topic solution of CTM reflects the relationship among topics in addition to the topic solution. The Table 6 shows the 10 topics solution of CTM. In the similar way in LDA, CTM also has adjective word and noun word for each topic with high probability, allowing easy labeling of topics. Moreover, CTM can identify relationship among topics. For example, we are able to make five similar topic groups using topic-document matrix showing the variational mean parameter for each document's topic proportions. The five similar topics are (0, 5), (1, 6), (2, 7), (3, 8), and (4, 9). For example, topic 0 and Topic 5 share camera, image, digital, and good words. Thus, CTM can be used to not only elicit topics but also to identify relationship among topics.

Table 6. The 10 topics solutions of CTM

T0:	T1:	T2:
camera(0.0513)[1]	camera(0.0552)	camera(0.0661)
image(0.0179)	pictures(0.0202)	just(0.0145)
great(0.0170)	great(0.0129)	pictures(0.0143)
powershot(0.0163)	easy(0.0122)	batteries(0.0132)
shoot(0.0161)	canon(0.0112)	shots(0.0112)
product(0.0145)	just(0.0110)	time(0.0098)
digital(0.0140)	good(0.0109)	good(0.0096)
good(0.0138)	photos(0.0105)	battery(0.0091)
like(0.0126)	batteries(0.0098)	mode(0.0090)
zoom(0.0125)	picture(0.0098)	great(0.0089)
http(0.0125)	like(0.0092)	using(0.0078)
optical(0.0123)	cameras(0.0082)	canon(0.0076)
href(0.0119)	digital(0.0076)	digital(0.0074)
amazon(0.0115)	clear(0.0075)	zoom(0.0073)
point(0.0113)	bought(0.0072)	bought(0.0067)
T3:	**T5:**	**T6:**
camera(0.0384)	camera(0.0516)	camera(0.0555)
batteries(0.0235)	canon(0.0221)	shots(0.0249)
card(0.0230)	image(0.0141)	battery(0.0194)
amazon(0.0114)	cameras(0.0136)	flash(0.0177)
memory(0.0110)	noise(0.0102)	batteries(0.0145)
battery(0.0101)	viewfinder(0.0100)	canon(0.0115)
like(0.0095)	digital(0.0096)	indication(0.0093)
just(0.0092)	shoot(0.0087)	card(0.0082)
download(0.0087)	pictures(0.0083)	time(0.0081)
option(0.0074)	quality(0.0082)	nimh(0.0075)
pictures(0.0073)	great(0.0075)	manual(0.0066)
chip(0.0069)	point(0.0073)	cameras(0.0060)
using(0.0066)	uses(0.0071)	turn(0.0060)
features(0.0065)	good(0.0070)	used(0.0060)
purchased(0.0062)	wanted(0.0066)	software(0.0059)

Overall, as a discriminitive model, LSA focuses on uniqueness in each topic, and can produce best result in the circumstance of single topic in a document. Generative models such as PLSA, LDA, and CTM emphasizes general trends in the collection of documents. PLSA can produce best result in single topic for one document, and LDA and CTM produce best result regardless of single or multiple topics for one document. Moreover, CTM can model the relationships between topics, and the relationships can be used to further grouping or to identify the structure of topics. The understanding of four methods in text mining and appropriate use of the methods not only help researchers apply proper methods in text mining in information systems research but also identify the advantages and limitations of each method.

5. Conclusion

In this paper, we have discussed the four methods in text mining including LSA, PLSA, LDA, and CTM. As a generative model, LSA can be used to seize unique or distinctive topics, and best applied in single topic document. Other three methods are generative models. In the generative models, the words having high marginal probabilities are appeared at the top of each topic. Because of general characteristic of the resultant topics, the generative methods can be used to seize overall or general trend in documents. In PLSA, it can be still applied in single topic of a document because it does not model at the document level. LDA overcomes this limitation using Dirichlet distribution, can be applied in each document having multiple topics. Furthermore, CTM adds the function to seize the relationships among topics, and it can be best utilized to seize the relationships among topics. The highlights of the strengths and weakness of each four methods help researchers apply appropriate methods in there research. We also expect that the introduction of text mining expands a broader audience within the IS research community. In particular, an experiment about topic detection provides a chance to compare four methods in text mining.

In sum, we believe that this paper contributes to literature by providing a comprehensive overview of the advantages and limitations of each of these various text mining methods that have been developed. Also, we believe that this paper contributes to literature by describing appropriate conditions to apply each method. We believe that this review can provide guidance of applying more refined methods such as generative methods in a host of IS research issues.

6. References

[1] D. Aldous, "Exchangeability and related topics", Ecole d'Ete de Probabilites de Saint-Flour XII, Springer Lecture Notes in Mathematics, 1117 (1985), pp. 1-198.

[2] M. Berry, *Survey of text mining: clustering, classification, and retrieval*, Springer-Verlag New York Inc, 2003.
[3] D. M. Blei and J. D. Lafferty, "A Correlated Topic Model of Science", The Annals of Applied Statistics, 1 (2007), pp. 17-35.
[4] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent Dirichlet Allocation", Journal of Machine Learning Research, 3 (2003), pp. 993-1022.
[5] K. Cheung, J. T. Kwok, M. H. Law and K. Tsui, "Mining customer product ratings for personalized marketing", Decision Support Systems, 35 (2003), pp. 231-243.
[6] R. J. Connor and J. E. Mosimann, "Concepts of Independence for Proportions with a Generation of the Dirichlet Distribution", Journal of the American Stastical Association, 64 (1969), pp. 194-206.
[7] S. R. Das and M. Y. Chen, "Yahoo! for Amazon: Sentiment extraction from small talk on the web", Management Science, 53 (2007), pp. 1375-1388.
[8] S. Deerwester, S. Dumais, G. Furnas, T. Landauer and R. Harshman, "Indexing by Latent Semantic Analysis", Journal of the American Society for Information Science, 41 (1990), pp. 391-407.
[9] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", Journal of the Royal Statistical Society. Series B (Methodological) (1977), pp. 1-38.
[10] C. H. Q. Ding, "A probabilistic model for Latent Semantic Indexing: Research Articles", Journal of the American Society for Information Science and Technology, 56 (2005), pp. 597-608.
[11] B. Fortuna, D. Mladenic and M. Grobelnik, "Semi-automatic construction of topic ontologies", Lecture Notes in Computer Science, 4289 (2006), pp. 121.
[12] N. Fuhr, "Probabilistic models in information retrieval", The Computer Journal, 35 (1992), pp. 243.
[13] M. Girolami and A. Kaban, *On an equivalence between PLSI and LDA*, ACM New York, NY, USA, 2003, pp. 433-434.
[14] T. L. Griffiths and M. Steyvers, *Finding scientific topics*, National Academy of Sciences, 2004, pp. 5228-5235.
[15] F. Haight, "Handbook of the Poisson distribution", New York (1967).
[16] D. Harman, *Overview of the first text retrieval conference (TREC-1)*, DIANE Publishing, 1993, pp. 1.
[17] T. Hofmann, *Probabilistic latent semantic indexing*, *SIGIR-99*, ACM New York, NY, USA, 1999, pp. 50-57.
[18] T. Hofmann, J. Puzicha and M. Jordan, "Unsupervised learning from dyadic data", Advances in Neural Information Processing Systems, 11 (1999).
[19] N. Ide and J. Veronis, "Introduction to the special issue on word sense disambiguation: the state of the art", Comput. Linguist., 24 (1998), pp. 2-40.
[20] R. Johson and D. Wichern, "Applied multivariate statistical analysis", Prentice Hall International (2001).
[21] T. K. Landauer, P. W. Foltz and D. Laham, "An introduction to latent semantic analysis", Discourse processes, 25 (1998), pp. 259-284.
[22] K. R. Larsen, D. E. Monarchi, D. S. Hovorka and C. N. Bailey, "Analyzing unstructured text data: using latent categorization to identify intelectual communities in information systems", Decision Support Systems, 45 (2008), pp. 884-896.
[23] P. Lenk, "The logistic normal distribution for Bayesian, nonparametric, predictive densities", Journal of the American Statistical Association (1988), pp. 509-516.
[24] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, Wiley New York, 1997.
[25] T. Miller, *Data And Text Mining: A Business Application Approach*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 2004.
[26] P. Pacheco, *Parallel programming with MPI*, Morgan Kaufmann, 1997.
[27] C. H. Papadimitriou, P. Raghavan, H. Tamaki and S. Vempala, "Latent semantic indexing: A probabilistic analysis", Journal of Computer and System Sciences, 61 (2000), pp. 217-235.
[28] A. Pons-Porrata, R. Berlanga-Llavori and J. Ruiz-Shulcloper, "Topic discovery based on text mining techniques", Information Processing and Management, 43 (2007), pp. 752-768.
[29] D. Rohde, "SVDLIBC", Available at http://tedlab.mit.edu/~dr/svdlibc.
[30] R. Rosenfeld, "Incorporating linguistic structure into statistical language models", Philosophical Transactions: Mathematical, Physical and Engineering Sciences (2000), pp. 1311-1324.
[31] R. Rosenfeld, "Two decades of statistical language modeling: where do we go fromhere?", Proceedings of the IEEE, 88 (2000), pp. 1270-1278.
[32] G. Salton, *Automatic text processing: the transformation, analysis, and retrieval of information by computer*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.
[33] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval", Information Processing and Management, 24 (1988), pp. 513-523.
[34] G. Salton, A. Wong and C. S. Yang, "A vector space model for automatic indexing", Communication sof the ACM, 18 (1975), pp. 613-620.
[35] A. Sidorova, N. Evangelopoulos, J. Valacich and T. Ramakrishnan, "Uncovering the intellectual core of the information systems discipline", MIS Quarterly, 32 (2008), pp. 467-482.
[36] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman and W. T. Freeman, *Discovering objects and their location in images*, *International Conference on Computer Vision (ICCV 2005)*, 2005.
[37] W. Strunk Jr, *The elements of style*, Filiquarian Publishing, LLC., 2007.
[38] P. C. Tetlock, M. Saar-Tsechansky and S. Macskassy, "More than words: Quantifying language to measure firms' fundamentals", Journal of Finance, 63 (2008), pp. 1437-1467.
[39] W. Yang and J. Dia, "Discovering cohesive subgroups from social networks for targeted advertising", Expert Systems with Applications, 34 (2008), pp. 2029-2038.