Improving the Performance of Wikipedia Based on the Entry Relationship between Articles

Lin-Chih Chen

Department of Information Management, National Dong Hwa University, Taiwan
lcchen@mail.ndhu.edu.tw

Abstract

Wikipedia is the largest online encyclopedia in the world. It is free to access by anyone and its main advantage is that it can also be edited by any person at any time. On the one hand, this caused a rapid growth to its number of available articles and languages. It is likely to cause that most users are difficult to differentiate various synonymy and polysemy terms from the millions of articles in Wikipedia. On the other hand, traditional semantic analysis models are mainly focus on to deal with the semantic relationships between terms, or terms and documents. However, these models are lacking to deal with the semantic relationships between documents.

In this paper, to enhance the semantic relationships between documents, we use the entry relationship between any two Wikipedia articles to design our Latent Entry Analysis (LEA) model. The advantages of LEA have the following several aspects: (1) it can effectively deal with the problems of synonymy and polysemy; (2) it is a good model to find the semantic relationships between terms, terms and documents, or documents; (3) it is a good model with a high-performance and low-cost compared to other semantic analysis models; (4) it is a suitable model to effectively handle big data sets in Wikipedia.

Keywords: Wikipedia articles, Entry relationship, Online internet encyclopedia, Semantic analysis models, Aspect model

1 Introduction

Wikipedia is an important part of today's Internet as it provides an entry point for finding information on a wide variety of topics such as mathematics, politics, literature, medicine, art, and computer science [36]. Wikipedia seeks to create a summary of all human knowledge in the form of an online encyclopedia, with each topic covered encyclopedically in one article. Since it has terabytes of disk space, it can have far more topics than can be covered by any printed encyclopedia [52].

The potential benefits of Wikipedia compared to traditional encyclopedias include at least the following: (1) it contains almost all possible topics in different subjects [12], (2) it allows a rapid response to any new introductions or events, and (3) it allows collaborative editing in the online encyclopedia [15].

Wikipedia is facing the problems of synonymy (two terms are syntactically different but semantically interchangeable expressions) and polysemy (a term has different meanings) that must be addressed. For example, for synonymy, if we talk about a long time or an extended time, long and extended are synonymous with that context. For polysemy, a very famous example is that "java" has at least two well-known meanings: it can be either a programming language or a location (Java island). These two problems are very important because most users are difficult to differentiate various synonym and polysemy terms from among the tens of million of Wikipedia entries [4].

Semantic analysis models are widely used to identify the semantic relationships between terms, or terms and documents [9, 23, 33-35]. In recent years, the most famous models are Latent Semantic Analysis (LSA) [26], Probabilistic LSA (PLSA) [18], and Latent Dirichlet Allocation (LDA) [2]. However, these models are lacking to deal with the semantic relationships between documents [45, 49]. This is also important because similar entries in Wikipedia always have the similar topic. To enhance the semantic relationships between documents, we propose a new model, called Latent Entry Analysis (LEA), to effectively identify the hidden semantic relationships between documents.

The main contributions of this paper are as follows. Firstly, to identify any possible semantic relationships between terms and documents, we propose a high-performance and low-cost LEA model. Secondly, we perform several experiments to verify the benefits of different semantic models and provide some suggestions for future directions for the research field of Wikipedia.

The rest of this paper is organized as follows. Firstly, in Section 2, we present a brief review of some previous literature relevant to this paper. Secondly, in

[*] Corresponding Author: Lin-Chih Chen; E-mail: lcchen@mail.ndhu.edu.tw

DOI: 10.3966/160792642018051903009

Section 3, we introduce all semantic analysis models used in this paper. Thirdly, in Section 4, we discuss about experiment results and discussion. Finally, in Section 5, we conclude this paper and discuss our future work.

2 Literature Review

In this section, we briefly review some research literature relevant to this paper, including Wikipedia applications and semantic analysis models. In this section, we provide two tables to compare relevant research in order to facilitate readers to read this literature.

2.1 Wikipedia Applications

There are many researchers who try to use Wikipedia as a primary source of research information to solve many different Information Retrieval (IR) problems. Table 1 shows the summary of some recent studies on Wikipedia applications. Milne and Witten [39] created a Wikipedia miner toolkit, which allows users to integrate Wikipedia's rich semantics into their applications. The benefit of using this toolkit is that it can effectively classify different elements of Wikipedia, such as article, label, and link by some classifiers. The classifiers used in this toolkit are the article and label comparers, the label and link disambiguators, and the link detector. Wu and Weld [53] proposed an open information extraction system to handle the unbounded number of relationships found on the Web by using heuristic matches between Wikipedia infobox attribute values and corresponding sentences in order to construct the training data set. Next, by the trained data set, the system can find the semantic relationships from natural language text. Lehmann et al. [28] proposed a DBpedia project to provide the query and search capabilities to the Wikipedia community. This project finally extracts some important structured data from Wikipedia by the following extractors: mapping-based infobox, raw infobox, feature and, statistical. Hahn et al. [14] proposed a faceted Wikipedia search mechanism to enable users to ask complex question against the Wikipedia knowledge base. This mechanism first sends the query to the DBpedia project to generate the facet values for each potential Wikipedia article, and then uses a sparse tree to store all facet values, which are used to answer the similar questions. Ciglan and Nørvåg [6] proposed a WikiPop system to detect significant increase of popularity of topics related to users' interests. This system first uses a Wikipedia link graph and graph-based recommendation algorithm to identify some related topics, and then uses the Wikipedia page view statistics to filter out the most popular related topics.

Table 1. The summary of Wikipedia applications

Researcher	Application	Approach	Advantage
Milne and Witten (2013)	Wikipedia miner toolkit	Article and label comparers, label and link disambiguators, and link detector	It can classify different elements of Wikipedia
Wu and Weld (2010)	Open information extraction system	A heuristic matches between Wikipedia attribute values	It can handle unbounded number of relationships
Lehmann et al. (2015)	DBpedia project	Mapping-based infobox, raw infobox, feature and statistical extractors	It provides the querying and search capabilities to a wide community
Hahn et al. (2010)	Faceted Wikipedia search	DBpedia & sparse tree	It allows a complex question in Wikipedia
Ciglan and Nørvåg (2010)	WikiPop system	Wikipedia link graph, graph algorithm, Wikipedia page view statistics	It can detect the most popular related topics in Wikipedia

2.2 Semantic Analysis Models

Currently, the best known semantic analysis models are LSA [26], PLSA [18], and LDA [2]. In this subsection, we briefly compare and analyze these models based on the concepts of approach, advantage and disadvantage, and problem-solving skills. Table 2 is the comparative study of different models.

LSA first uses Singular Value Decomposition (SVD) to separate the topic matrix from the original term-by-document matrix, and then uses a dimension-reduction technique to filter out the noisy topics from the topic matrix [44]. The advantage of LSA is that it can handle the problem of identifying synonymy by using the dimension-reduction technique [8]. In contrast, the disadvantage of LSA is that it cannot effectively deal with the problem of identifying polysemy because SVD is a one-to-one mapping technique from a particular term to a particular document [27]. There are many researchers who used LSA to solve different IR problems. For the assessment of short free text answer, there have been some researchers [24, 32] who tried to use LSA to automatically assess whether the answer is correct. For the problem of document summarization in a large number of user-generated documents, some researchers [42, 55] used LSA to sort all documents to

Table 2. The comparative results for different semantic analysis models

Semantic Analysis Models	Approach	Advantage (A) and Disadvantage (D)	Problem Solving
LSA	SVD and Dimension-reduction	A: Synonymy D: Polysemy	Assessment of free text answer, Document summarization, Plagiarism detection in source code
PLSA	EM algorithm	A: Synonymy and Polysemy D: Huge Computing time	Predicting the mobility patterns, Sports video summarization, Detecting soung events of human activities
LDA	Dirchlet probability distribution and Gibbs sampling algorithm	A: an unsupervised generative model D: Not suitable for a small amount data or normal distribution	Automatically classify a large number of bug reports, Recommending relevant multimedia tags, Automatically annotate the image

achieve the purpose of document summarization. For the problem of plagiarism detection in source code, Cosma and Joy [7] tried to use LSA to decide whether a piece of source code is plagiarism.

PLSA uses an iterative Expectation-Maximization (EM) algorithm to estimate the latent probabilities in the term-by-document matrix. The EM algorithm first calculates the expectation value of latent topics based on the probability of the observation parameters, and then updates the probability of the observation parameters by maximizing the objective function. Compared to LSA, the advantage of PLSA is that it can further handle the problem of identifying polysemy because the EM algorithm is a statistical estimation technique that can estimate multiple parameters simultaneously [10, 46]. Conversely, the disadvantage of PLSA is that it needs a huge computation time because the EM algorithm is a time-consuming algorithm [3, 21, 29]. In the past, many researchers used PLSA to solve various IR problems. To improve the location prediction services, McInerney et al. [37] used PLSA to predict the mobility patterns for some new users. For online sports video summarization and retrieval, Xu et al. [54] first analyzed a text derived from the broadcast video and then used PLSA to detect all possible sport events. Mesaros et al. [38] used a two-stages approach to detect specific sound events of human's daily activities. In the first stage, PLSA is used to find the relationships between events. In the second stage, PLSA is used again to update the probabilities of events according to the history of events.

LDA first uses the Dirichlet probability distribution to set up the latent probability distributions of topics, terms, and documents; and then uses the Gibbs sampling algorithm to estimate the latent probabilities of topics and terms for a given document [31]. The advantage of LDA is that it can easily find all corresponding terms for each topic because LDA is based on an unsupervised generative model without any prior information about topics and terms [1]. The disadvantage of LDA is that it is not suitable for the environment of a small amount data or normal distribution [51]. There are many researchers who used LDA to solve different IR problems. To automatically classify a large number of bug reports, Somasundaram and Murphy [47] used LDA to help the user to streamline the process of solving the bug. To help the user to search relevant multimedia content, Krestel et al. [25] used LDA to recommend relevant multimedia tags to user. For implementing an efficient image annotation method, Liénou et al. [30] first searched all pattern relationships between different image frames, and then applied LDA to automatically annotate the image according to the pattern relationships.

3 Research Method

In this section, we briefly describe our system architecture as shown in Figure 1. Our system mainly involves two stages: Data Preprocessing and Data Calculating. In the first stage, Data Processing, to obtain the data source of our study, we first use a Web Spider to crawl the Search Engine Results Pages (SERPs) returned from Wikipedia. Since the result of SERP is an unstructured document, we then use the Perl Compatible Regular Expressions (PCRE) and some Natural Language Processing (NLP) techniques to convert unstructured SERP documents into structured documents. We last use the Term Frequency Inverse Document Frequency (TFIDF) method to convert structured documents into a term-by-document matrix because the input of semantic analysis modes is a matrix form. In the second stage, Data Calculating, we first use the LSA, PLSA, and LDA models as the benchmark of our study, and then develop the Latent Entry Analysis (LEA) model to strengthen the entry relationship between Wikipedia articles.

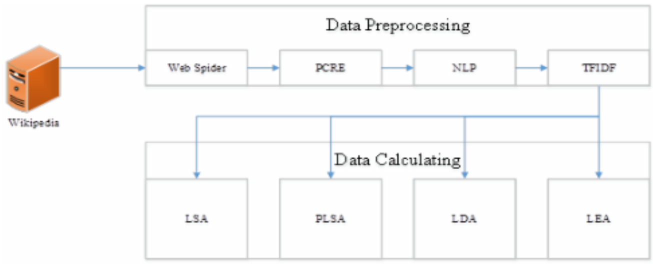


Figure 1. The flow chart of our study

3.1 Data Preprocessing Stage

In this stage, we first develop a Web Spider technique to simultaneously crawl the SERPs returned from Wikipedia. In theory, by our technique, the crawling time for multiple SERPs is same as single SERP when the network latency and bandwidth are not considered.

Since the SERP is an unstructured document [11, 40], we next use the PCRE technique [16] to do the pattern match on all collected unstructured documents to draw out some important analysis elements, such as page's title, URL, and description.

We then use some NLP techniques to prevent unnecessary words appeared in the structured documents. In this study, we use the following NLP techniques: stemming, stop words, non-words. To prevent a word has different forms but essentially the same meaning, we use the Porter stemming algorithm [43] to convert the word into its root word. To prevent common words appeared in the structured documents, we use Google's suggestion [13], which contains 671 stop words, such as articles, prepositions, and pronouns, to filter out these common words. To prevent unnecessary characters appeared in the structured documents, we strip out all non-words characters, such as punctuations, special-characters, space, and HTML tags.

We last use the TFIDF method to transform structured documents into a term-by-document matrix, which is the input of semantic analysis models. The transformation process is shown in the following equation.

$$TD(d,t)=\frac{N(d,t)}{N(d)}\log(\frac{|D|}{df(t)}) \qquad \textbf{(1)}$$

where $TD(d,t)$ represents the weight of a document d containing the term t; $N(d,t)$ represents the number of occurrence of document d containing term t; $N(d)$ represents the total number of terms in document d; $|D|$ represents the number of documents in our structured documents; $df(t)$ is the number of documents where term t occurs.

3.2 Data Calculating Stage

In this stage, we first discuss the concepts of LSA, PLSA and LDA, and then propose and illustrate our LEA model in detail.

LSA. LSA first uses the SVD technique, as shown in the following equation, to decompose the original term-by-document matrix into three new matrices; where S is the topical semantic space, T is the term space corresponding to S, D is the document space corresponding to S.

$$TD(d,t)=(d,t)=TSD' \qquad \textbf{(2)}$$

LSA then uses the dimension-reduction technique to filter out the noisy topics from S. To obtain a noisy-free semantic space S', we retain the K largest eigenvalues from S and other eigenvalues are treated as noise. LSA last calculates a new term-by-document matrix with the noisy-free semantic space by the product of T, S' and D' matrices.

PLSA. PLSA uses the aspect model [17], which is a latent variable model for co-occurrence data associating an unobserved class variable, to estimate the latent probabilities between terms and documents. The aspect model is shown in Figure 2. In this model, PLSA uses an iterative EM algorithm and the maximum likelihood estimation method to estimate the latent probability of a document containing the term.

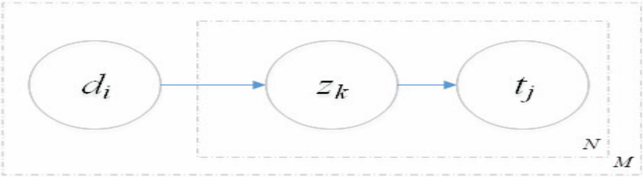


Figure 2. The aspect model of PLSA

To model the aspect model, PLSA selects a particular topic z_k on a given document d_i, and a particular term t_j on a given topic z_k. The notations of PLSA are defined as follows: M is the number of documents, N is the number of terms, $d_i \in \{d_1,\ldots,d_M\}$ is a particular document, $t_j \in \{t_1,\ldots,t_N\}$ is a particular term, $z_k \in \{z_1,\ldots,z_K\}$ represents an observed topic variable.

Based on the aspect model, PLSA estimates the latent probability of a document d_i containing the term t_j, $p(d_i,t_j)$, by the following equation; where $p(z_k)$ is the probability of topic z_k, and $p(X|Y)$, X and $Y \in \{d_i, t_j, z_k\}$, is the conditional probability of X given that Y occurs.

$$p(d_i,t_j)=\sum_k p(t_j\,|\,z_k)p(z_k)p(d_i\,|\,z_k) \qquad \textbf{(3)}$$

PLSA uses the iterative EM algorithm and the Maximum Likelihood Estimation (MLE) method to calculate the final result of $p(d_i,t_j)$. The detailed process of the derived PLSA parameters is shown below.

PLSA first defines the following likelihood function as the objective function of PLSA by applying the MLE method; where $n(d_i,t_j)$ denotes the weight of document d_i containing t_j.

$$L(d_i,t_j)=\sum_i\sum_j n(d_i,t_j)\log p(d_i,t_j) \tag{4}$$

PLSA then uses the EM algorithm to maximize the objective function. The EM algorithm repeatedly executes the Expectation (E) and Maximization (M) steps until the termination condition is reached. The EM algorithm first estimates the expectation of the hidden topic based on the probabilities of the observation parameters by the E step, and then updates the probabilities of the observation parameters by the M step. In the E step, the EM algorithm uses the probabilities of the observed parameters to estimate the expectation of hidden topic parameter, as shown in the following equation.

$$p(z_k\mid d_i,t_j)=\frac{p(d_i\mid z_k)p(z_k)p(t_j\mid z_k)}{\sum_k p(d_i\mid z_k)p(z_k)p(t_j\mid z_k)} \tag{5}$$

In the M step, the EM algorithm first uses the result of E step to maximize the log likelihood of the objective function, and then updates the probabilities of the observed parameters in equation 3, as shown in the following equations.

$$p(t_j\mid z_k)=\frac{\sum_j n(d_i\mid t_j)p(z_k\mid d_i,t_j)}{\sum_j\sum_i n(d_i\mid t_j)p(z_k\mid d_i,t_j)} \tag{6}$$

$$p(z_k)=\frac{\sum_j\sum_i n(d_i\mid t_j)p(z_k\mid d_i,t_j)}{\sum_j\sum_i n(d_i\mid t_j)} \tag{7}$$

$$p(d_i\mid z_k)=\frac{\sum_i n(d_i\mid t_j)p(z_k\mid d_i,t_j)}{\sum_j\sum_i n(d_i\mid t_j)p(z_k\mid d_i,t_j)} \tag{8}$$

LDA. LDA use the Dirichlet probability distribution to estimate the probability distribution between terms and documents as shown in Figure 3. Compared to LSA, the advantage of LDA is that it is a probabilistic model with interpretable topics. In the LDA model, α denotes the parameters of the Dirichlet distribution prior on the topic-by-document distributions, β denotes the parameters of the Dirichlet distribution prior on the term-by-topic distributions, K is the number of topics, M is the number of documents, N is the number of terms, θ_i denotes the topic distribution of document i which is corresponding to the conjugate distribution of parameter α in the Dirichlet distribution, φ_z denotes the terms distribution of topic z which is corresponding to the conjugate distribution of parameter β in the Dirichlet distribution, $z_{i,j}$ denotes the topic of document i containing term j which is corresponding to the conjugate distribution of parameter θ_i in the multinomial distribution, and t_i denotes the specific term in document i which is corresponding to the conjugate distribution of parameter φ_z in the multinomial distribution.

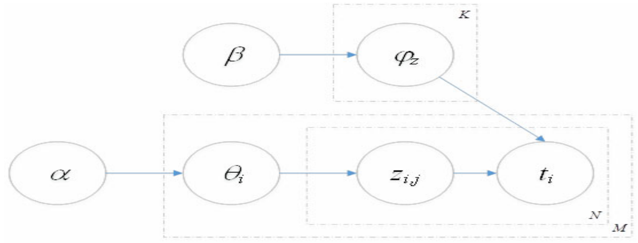


Figure 3. The aspect model of LDA

LDA first estimates the conditional probability of a K dimensional Dirichlet variable θ, given the parameter α ($\theta_k \geq 0$ and $\alpha_k \geq 0$), as shown in the following equation; where α is a K-vector with elements α_k, $\Gamma(x)$ is the Gamma function.

$$p(\theta\mid\alpha)=\frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)}\theta_1^{\alpha_1-1}\cdots\theta_K^{\alpha_K-1} \tag{9}$$

LDA then estimates the joint distribution probability of all observed and unobserved parameters given the partial observed parameters, α and β, as shown in the following equation.

$$\begin{aligned}&(t_i,z_i,\theta_i,\varphi_z\mid\alpha,\beta)\\&=\prod_j p(\theta_i\mid\alpha)p(z_{i,j}\mid\theta_i)p(t_i\mid\varphi_z)p(\varphi_z\mid\beta)\end{aligned} \tag{10}$$

LDA last estimates the MLE of the terms distribution of a document by integrating over θ_i and summing over φ_z, as shown in the following equation.

$$\begin{aligned}&p(t_i\mid\alpha,\beta)\\&=\int_{\theta_i}\int_{\varphi_z} p(t_i,z_i,\theta_i,\varphi_z\mid\alpha,\beta)d\varphi_z d\theta_i\end{aligned} \tag{11}$$

In summary, LDA first uses the Dirichlet and multinomial distributions to set up the relevant parameters, and then uses the Gibbs sampling algorithm [48] to estimate topics from the collected documents as well as estimate the term-by-topic and topic-by-document probabilities. Compared to PLSA, the advantage of LDA is that it can significantly reduce the computation time because it uses the Gibbs sampling algorithm rather than the EM algorithm to estimate the probabilities of different parameters [48].

LEA. The above three semantic analysis models can effectively identify the semantic relationships between terms, or terms and documents; but they are lacking to find the semantic relationships between documents. This is important because similar Wikipedia articles always have the similar topic. To address this problem, we propose a new model, called LEA, to emphasize the entry relationship between Wikipedia articles.

LEA first needs to find a model to describe the terms and topics relationships on a particular document. According to LDA's description, it is a suitable model to estimate the term-by-topic and topic-by-document probabilities for a given document. Thus, to build the document-link relationship between documents, we need two LDA models, LDA_x and LDA_y, to represent the LDA models of two different documents, x and y. To connect these two LDA models, we additionally include an entry relationship parameter η and generate a binary link variable c in our LEA model. Figure 4 is the LEA model with the entry relationship between documents x and y. In the LEA model, η is the parameters of the Dirichlet distribution prior on the topic-by-link distributions, c_e denotes the topic distribution of entry e which is corresponding to the conjugate distribution of parameter η in the Dirichlet distribution, E denotes the number of entry relationships.

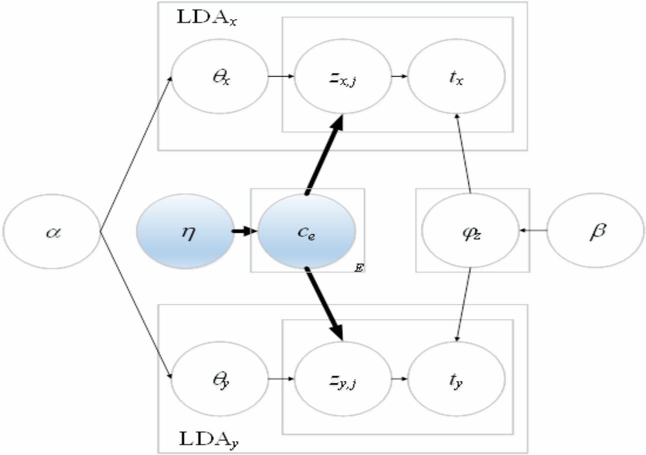


Figure 4. The aspect model of LEA

For any pair of documents, LEA uses the topic-by-entry relationship η between two documents to estimate that a topic how to affect the entry relationship. For all collected documents, LEA uses the weighted average of all ηs to estimate that different topics how to affect the entry relationship.

Let us now discuss the estimation equation of LEA model in detail. LEA first estimates the joint distribution probability of all observed and unobserved parameters given the partial observed parameters, α, β, and η, as shown in the following equation, where $i \in \{x, y\}$.

$$p(t_i,z_i,\theta_i,\varphi_z,c_e \mid \alpha,\beta,\eta) \\ = \prod_{j}\prod\prod_{i} p(\theta_i \mid \alpha)p(z_{i,j} \mid \theta_i)p(t_i \mid \varphi_z)p(z_{i,j} \mid c_e)p(\varphi_z \mid \beta)p(c_l \mid \eta) \quad \textbf{(12)}$$

LEA then estimates the MLE of the terms distribution of all documents by summing over θ_i, φ_z, and c_e, as shown in the following equation.

$$p(t_i \mid \alpha,\beta,\eta) \\ = \int_{\theta_i}\int_{\varphi_z}\int_{\varphi_z} p(t_i,z_i,\theta_i,\varphi_z,c_e \mid \alpha,\beta,\eta)\, dc_e d\varphi_z d\theta_i \quad \textbf{(13)}$$

Table 3 is the comparison results of different semantic analysis models. LSA uses the SVD and dimensional reduction techniques to filter out the noisy topics from the original data. By the dimensional reduction technique, it can effectively identify the synonymy between terms. PLSA uses the EM algorithm and MLE method to estimate the relevant parameters. It can further identify the polysemy between terms because the EM algorithm can simultaneously estimate multiple parameters. LDA and LEA use the Gibbs sampling algorithm and MLE method to estimate the relevant parameters. The difference of these two models is that they have different focus points. In LDA, it focuses on the topics between terms because two parameters α (topic-by-document) and β (term-by-topic) are focused on the process of topics. In LEA, it focuses on how to effectively identify the entry relationship between documents. By LEA, we can easily cluster the similar documents with the similar topic into a cluster. For the consideration of the speed of computation, LSA is the fastest because it does not use an iterative method to estimate the parameters. Conversely, PLSA is the slowest because the computation time of the EM algorithm is increasing exponentially along increasing the number of terms and documents [5]. The speed of LDA and LEA is significantly faster than PLSA because these two models use the Gibbs sampling algorithm rather than the EM algorithm to estimate the parameters. In the next section, we will present some experiments to compare the performance and cost (the computation time) of different semantic analysis models.

4 Experimental Results and Analysis

In this experiment, the experimental data is selected from top 100 viewed entries on Wikipedia from years 2012 to 2014. By screening all entries, we delete some repeating entries and retain only one entry. The view count of each entry is the sum of its view counts for each year. Table 4 is the top 20 most popular entries on Wikipedia. In this study, we selected a total of 206 entries as our experimental data. The full set of experimental data is shown in http://goo.gl/wn5aHZ.

Table 3. The comparison of different semantic analysis models

Model	Calculation Method Estimation Equation	Speed	Focus Point
LSA	SVD & Dimensional Reduction $TD(d,t) = TSD'$	Fastest	Symonymy between Terms
PLSA	EM & MLE $p(d_i,t_j) = \sum_k p(t_j \mid z_k) p(z_k) p(d_i \mid z_k)$	Slowest	Polysemy between Terms
LDA	Gibbs & MLE $p(t_i,z_i,\theta_i,\varphi_z \mid \alpha,\beta) = \prod_j p(\theta_i \mid \alpha) p(z_{i,j} \mid \theta_i) p(t_i \mid \varphi_z) p(\varphi_z \mid \beta)$ $p(t_i \mid \alpha,\beta) = \int_{\theta_i}\int_{\varphi_z} p(t_i,z_i,\theta_i,\varphi_z \mid \alpha,\beta) d\varphi_z d\theta_i$	Fast	Topics between Terms
LEA	Gibbs & MLE $p(t_i,z_i,\theta_i,\varphi_z,c_e \mid \alpha,\beta,\eta) = \prod_j \prod_i p(\theta_i \mid \alpha) p(z_{i,j} \mid \theta_i) p(t_i \mid \varphi_z) p(z_{i,j} \mid c_e) p(\varphi_z \mid \beta) p(c_l \mid \eta)$ $p(t_i \mid \alpha,\beta,\eta) = \int_{\theta_i}\int_{\varphi_z}\int_{c_e} p(t_i,z_i,\theta_i,\varphi_z,c_e \mid \alpha,\beta,\eta) dc_e d\varphi_z d\theta_i$	Fast	Entry Relationship between documents

Table 4. The partial list of English Wikipedia's most popular entries from 2012 to 2014

Entry	Year Appears	View Count
Facebook	2012, 2013, 2014	83295511
Wiki	2012, 2013, 2014	56352325
United States	2012, 2013, 2014	45700980
YouTube	2012, 2013, 2014	44482179
Java	2013, 2014	43172982
The Walking Dead (TV Series)	2012, 2013, 2014	39762769
Wikipedia	2012, 2013, 2014	36890189
The Big Bang Theory	2012, 2013, 2014	36887175
Google	2012, 2013, 2014	35439215
Breaking Bad	2012, 2013, 2014	35360135
World War II	2012, 2013, 2014	35229756
Online shopping	2014	34897548
Climatic Research United email controversy	2013, 2014	34025774
Alive	2013, 2014	33257949
How I Met Your Mother	2012, 2013, 2014	32741176
India	2012, 2013, 2014	32627605
One Direction	2012, 2013	32152677
Sex	2012, 2013, 2014	30113283
Game of Thrones	2013, 2014	28717418
Eminem	2012, 2013, 2014	28531162

For the data preprocessing stage in Figure 1, we use the PHP scripting language to generate the term-by-document matrix crawled from Wikipedia; and for the data calculating stage, we use the MATLAB programming language to simulate the results of different semantic analysis models. To compare the performance of different models, we need some similarity functions to evaluate the similarity measure between two terms or documents vectors in a matrix because the result of models is a matrix form. The similarity function can be any vector similarity function. The most commonly used functions are the COSine similarity (COS) and CORrelation coefficient (COR) [19, 41, 50]. Thus, in this study, we also use these two similarity functions to compare the performance of different models.

The number of topics will influence the performance of different models. On the one hand, a larger number of topics may result in the performance being degradation. On the other hand, a small number of topics may result in the document being ambiguous. Thus, how to find the appropriate topic number correctly is a very important problem. According to the suggestion of Hofmann et al. [20], we set the number of topics range from 5 to 50 to evaluate the performance of different models.

We first input two queries "JB" and "LSA" to Wikipedia to carefully detail and explain the problems of synonymy and polysemy how to impact the performance of different models. Tables 5-1 and 5-2 are the results of "JB" and "LSA" for different models. According to these two tables, we find that the terms, "JB" and "James Bond" (or "James Brown"), have the feature of synonymy. Similarly, the terms, "LSA" and "Late Stone Age" (or "Latent Semantic Analysis"), also have the feature of synonymy; that is, the abbreviation of "Late Stone Age" or "Latent Semantic Analysis" is "LSA". Moreover, the term "JB" is also a polysemous term because it has at least six distinct meanings, "James Bond", "James Brown", "Joe Biden", "Junction Box", "Jailbreaking", and "Jelly bean". Looking again at these two tables, for all models except LSA, the scores of COS and COR between

Table 5-1. An example of the results of cosine (COS) and correlation coefficient (COR) for different models for the query is "JB"

LSA				PLSA			
Document / Term	d_1	d_2	d_3	Document / Term	d_1	d_2	d_3
James Bond	0.16	0.56	0.62	James Bond	0.13	0.71	0.77
James Brown	0.68	0.12	0.46	James Brown	0.73	0.22	0.23
Joe Biden	0.13	0.95	0.12	Joe Biden	0.08	0.88	0.03
Junction Box	0.02	0.12	0.56	Junction Box	0.05	0.62	0.16
Jailbreaking	0.05	0.82	0.13	Jailbreaking	0.22	0.42	0.08
Jelly Bean	0.19	0.97	0.12	Jelly Bean	0.07	0.87	0.10
	COS	COR			COS	COR	
Between Terms	0.62563	0.02316		Between Terms	0.74445	0.28356	
Between Documents	0.51884	0.65071		Between Documents	0.49691	0.72293	
LDA				LEA			
Document / Term	d_1	d_2	d_3	Document / Term	d_1	d_2	d_3
James Bond	0.58	0.05	0.37	James Bond	0.50	0.74	0.45
James Brown	0.22	0.77	0.23	James Brown	0.31	0.95	0.21
Joe Biden	0.95	0.13	0.13	Joe Biden	0.35	0.43	0.42
Junction Box	0.32	0.02	0.16	Junction Box	0.04	0.16	0.63
Jailbreaking	0.82	0.05	0.08	Jailbreaking	0.03	0.16	0.52
Jelly Bean	0.57	0.08	0.68	Jelly Bean	0.50	0.62	0.53
	COS	COR			COS	COR	
Between Terms	0.73819	0.25816		Between Terms	0.79399	0.32949	
Between Documents	0.48206	0.72775		Between Documents	0.75181	0.86766	

Table 5-2. An example of the results of cosine (COS) and correlation coefficient (COR) for different models for the query is "LSA"

LSA				PLSA			
Document / Term	d_1	d_2	d_3	Document / Term	d_1	d_2	d_3
Late Stone Age	0.24	0.46	0.59	Late Stone Age	0.09	0.61	0.71
Latent Semantic Analysis	0.70	0.19	0.41	Latent Semantic Analysis	0.72	0.25	0.15
Light Small Arms	0.23	0.90	0.17	Light Small Arms	0.10	0.77	0.10
Legal Services Agency	0.08	0.08	0.50	Legal Services Agency	0.02	0.53	0.11
London Stansted Airport	0.03	0.85	0.04	London Stansted Airport	0.24	0.46	0.14
Light-Sport Aircraft	0.12	0.93	0.07	Light-Sport Aircraft	0.05	0.90	0.06
	COS	COR			COS	COR	
Between Terms	0.61848	0.02145		Between Terms	0.74420	0.26456	
Between Documents	0.54494	0.63124		Between Documents	0.47012	0.69163	
LDA				LEA			
Document / Term	d_1	d_2	d_3	Document / Term	d_1	d_2	d_3
Late Stone Age	0.53	0.05	0.37	Late Stone Age	0.55	0.70	0.41
Latent Semantic Analysis	0.25	0.77	0.23	Latent Semantic Analysis	0.35	0.90	0.20
Light Small Arms	0.9	0.13	0.13	Light Small Arms	0.30	0.41	0.39
Legal Services Agency	0.27	0.02	0.16	Legal Services Agency	0.14	0.20	0.53
London Stansted Airport	0.77	0.05	0.08	London Stansted Airport	0.23	0.10	0.42
Light-Sport Aircraft	0.6	0.08	0.68	Light-Sport Aircraft	0.41	0.52	0.31
	COS	COR			COS	COR	
Between Terms	0.75063	0.26177		Between Terms	0.83938	0.33134	
Between Documents	0.4998	0.74133		Between Documents	0.82091	0.85277	

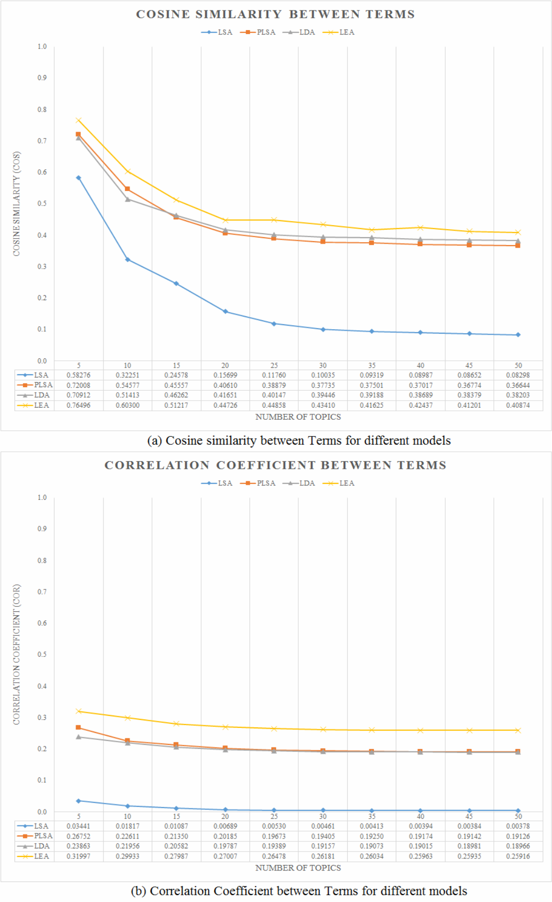
terms or documents are significantly better than LSA. This implies that all models can effectively identify the semantic relationships between different synonymous terms, but LSA lacks to identify the semantic relationships between different polysemous terms. That is, all models can deal with the problem of synonymy, but LSA lacks to deal with the problem of polysemy. This result also echoed the finding of Ishida and Ohata [22]. That is, LSA can effectively deal with the problem of synonymy, but it lacks the capability to deal with the problem of polysemy; because by using SVD technique, a row vector in a matrix can only represent a term. Other models use the aspect model to simultaneously estimate the joint probability of terms and documents. By using the aspect model, it can clearly distinguish different meanings and types between terms so that the model can gracefully deal with the problem of polysemy.

Next, we conduct an extended experiment on a large-scale dataset. For the above-mentioned 206 entries, we first crawl the Wikipedia articles and then perform the data preprocessing stage in Figure 1 to generate the term-by-document matrix.

Figure 5 is the results of the COS and COR between terms for different models. Each dot in the figure is the average value of all different articles for a specific number of topics. According to the results of the figure, the COS and COR are decreased along with the number of topics is increased. The best and worst models are LEA and LSA, respectively. The COS for LSA is a clear downward trend when the number of topics is greater than 5, but other models are less obvious. This implies that LSA is not suitable to handle multiple topics at the same time. Although the COS and COR of PLSA are similar to LDA, it needs a huge amount of computation time to reach the final solution, as described later in detail. This implies that PLSA is not a suitable model to handle big data sets. The base model of LEA is LDA; thus, LEA can significantly reduce the computation time as LDA compared to PLSA. This implies that LEA and LDA are two suitable models to handle big data sets. Compared to LDA and LEA, the COS and COR of LEA are better than LDA. This implies that when the entry relationship is applied to LDA, the performance of LDA can be effectively improved.

Figure 6 is the results of the COS and COR between documents for different models. According to the results of the figure, all models except LEA have similar lower performance; that is, these models cannot effectively identify the semantic relationships between documents. LEA uses the entry relationship between documents to effectively identify the semantic relationships between documents. This implies that we can easily cluster the similar documents with the similar topic into a cluster by LEA. In summary, LEA is a good model to find the semantic relationships between terms, terms and documents, or documents.

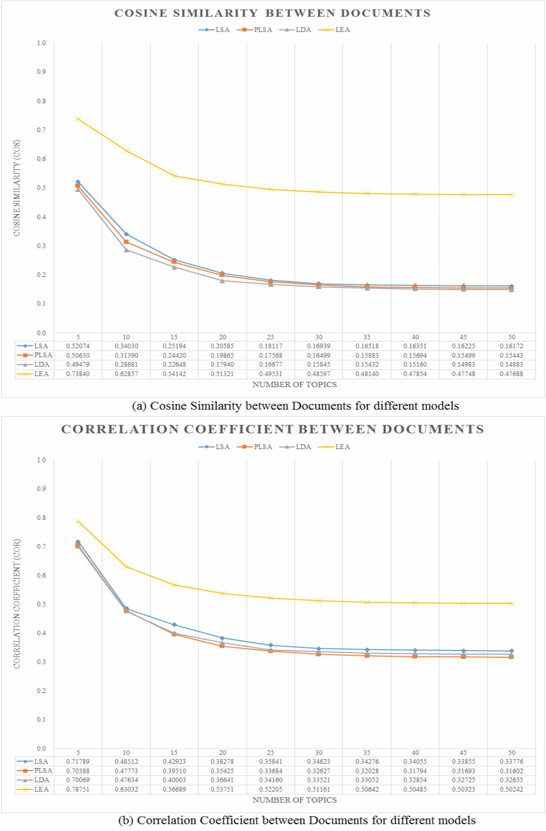
We then use SPSS 14.0 for Windows to analyze the results of above two experiments. Considering the COS and COR of different models, we use the statistical methodology, Analysis of Variance (ANOVA) analysis, to show that $F_{(K=5)}$=112.871, $F_{(K=10)}$=430.771, $F_{(K=15)}$=555.322, $F_{(K=20)}$=723.694, $F_{(K=25)}$=884.042, $F_{(K=30)}$=903.743, $F_{(K=35)}$=901.851, $F_{(K=40)}$=929.021, $F_{(K=45)}$=928.332, $F_{(K=50)}$=932.991 (Table 6) are all greater than $F_{0.001}(3,3292)$=5.435 (F-distribution). This provides extremely strong evidence against the null hypothesis, indicating that there is a significant difference in the COS and COR of different models.



(a) Cosine similarity between Terms for different models

(b) Correlation Coefficient between Terms for different models

Figure 5. The COS and COR between terms for different models



(a) Cosine Similarity between Documents for different models

(b) Correlation Coefficient between Documents for different models

Figure 6. The COS and COR between documents for different models

Table 6. The result of ANOVA analysis

		Sum of Squares	df	Mean Square	F	Sig.
$K = 5$	Between Groups	15.025	3	5.008	112.871	.000
	Within Groups	146.070	3292	0.44		
	Total	161.095	3295			
$K = 10$	Between Groups	26.561	3	8.854	430.71	.000
	Within Groups	67.661	3292	.021		
	Total	94.222	3295			
$K = 15$	Between Groups	24.790	3	8.263	555.322	.000
	Within Groups	48.985	3292	.015		
	Total	73.774	3295			
$K = 20$	Between Groups	26.917	3	8.972	723.694	.000
	Within Groups	40.814	3292	.012		
	Total	67.731	3295			
$K = 25$	Between Groups	30.056	3	10.019	884.042	.000
	Within Groups	37.308	3292	.011		
	Total	67.364	3295			
$K = 30$	Between Groups	29.383	3	9.794	903.743	.000
	Within Groups	35.677	3292	.011		
	Total	65.059	3295			
$K = 35$	Between Groups	29.377	3	9.792	901.851	.000
	Within Groups	35.745	3292	.011		
	Total	65.122	3295			
$K = 40$	Between Groups	29.669	3	9.890	929.021	.000
	Within Groups	35.044	3292	.011		
	Total	64.713	3295			
$K = 45$	Between Groups	29.323	3	9.774	928.332	.000
	Within Groups	34.662	3292	.011		
	Total	63.985	3295			
$K = 50$	Between Groups	29.428	3	9.809	932.991	.000
	Within Groups	34.611	3292	.011		
	Total	64.039	3295			

We conducted a post hoc Fisher's Least Significant Difference (LSD) for pair-wise comparison at the 1% significance level. Because the results of LSD are tedious, we refer readers to the full report at http://goo.gl/EGjJfC. As illustrated in the results of LSD, LEA (LSA) was found to overwhelmingly better (worse) than other models.

Figure 7 is the computation time of different models. The computation time of LSA is fastest compared to other models. This is because that it does not uses an iterative method to reach the final solution. However, its performance is worst because it only deal with the problem of synonymy. According to the results of the figure, PLSA totally needs 142815.3 second to run the EM algorithm. This implies that PLSA needs a huge amount of computation time to reach the final solution. This is because that the computation time of the EM algorithm is increasing exponentially along increasing the number of terms and documents. Compared to PLSA, the advantage of LDA and LEA is that they can significantly reduce the computation time because they use the Gibbs sampling algorithm rather than the EM algorithm to reach the final solution.

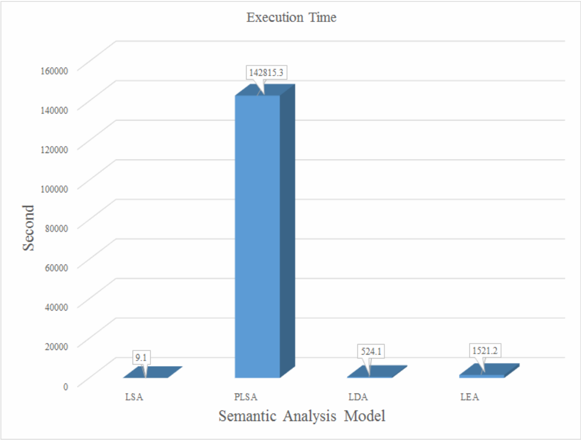


Figure 7. The computing time for different models

5 Conclusion and Future Work

In this paper, we use the entry relationship between Wikipedia articles to design our semantic model. The one advantage of our model is that it can effectively identify the semantic relationships between terms, terms and documents, or documents. The performance of our model is significantly better than other models. The other advantage of our model is that it can significantly reduce the computation time so that our model is suitable to handle big data sets in Wikipedia.

In the future, we plan to add a time series relationship to our model to further cluster the similar documents with the similar updated time into a cluster. By this new relationship, we can clearly distinguish between iPhone 6 and 6S because these two types of iPhone mobile phones have different released dates.

Acknowledgements

We would like to thank anonymous reviewers of the paper for their constructive comments which help us to improve the paper in serveral ways. This work was supported in part by Ministry of Science and Technology, Taiwan under Grant Most 106-2410-H-259-011, 105-2221-E-259-030 & 104-2221-E-259-038.

References

[1] D. M. Blei, Probabilistic Topic Models, *Communications of the ACM*, Vol. 55, No. 4, pp. 77-84, April, 2012.

[2] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet Allocation, *Journal of Machine Learning Research*, Vol. 3, No. 1, pp. 993-1022, January, 2003.

[3] A. V. Brahmane, A. Amune, A Survey of Dynamic Distributed Network Intrusion Detection Using Online Adaboost-Based Parameterized Methods, *International Journal of Innovative Research in Advanced Engineering*, Vol. 1, No. 9, pp. 256-262, October, 2014.

[4] CBS, *Wikipedia Cofounder Jimmy Wales on 60 Minutes*, http://www.cbsnews.com/news/wikipedia-jimmy-wales-morley-safer-60-minutes/.

[5] L.-C. Chen, Building a Term Suggestion and Ranking System Based on a Probabilistic Analysis Model and a Semantic Analysis Graph, *Decision Support Systems*, Vol. 53, No. 1, pp. 257-266, April, 2012.

[6] M. Ciglan, K. Nørvåg, Wikipop- Personalized Event Detection System Based on Wikipedia Page View Statistics, *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, Toronto, Canada, 2010, pp. 1931-1932.

[7] G. Cosma, M. Joy, An Approach to Source-Code Plagiarism Detection and Investigation Using Latent Semantic Analysis, *IEEE Transactions on Computers*, Vol. 61, No. 3, pp. 379-394, March, 2012.

[8] S. T. Dumais, Latent Semantic Analysis, *Annual Review of Information Science and Technology*, Vol. 38, No. 1, pp. 188-230, 2004.

[9] O. Egozi, S. Markovitch, E. Gabrilovich, Concept-Based Information Retrieval Using Explicit Semantic Analysis, *ACM Transactions on Information Systems*, Vol. 29, No. 2, pp. 8:1-8:34, April, 2011.

[10] R. Fernandez-Beltran, F. Pla, Incremental Probabilistic Latent Semantic Analysis for Video Retrieval, *Image and Vision Computing*, Vol. 38, No. C, pp. 1-12, June, 2015.

[11] M. Gärtner, A. Rauber, H. Berger, Bridging Structured and Unstructured Data Via Hybrid Semantic Search and Interactive Ontology-Enhanced Query Formulation, *Knowledge and Information Systems*, Vol. 41, No. 3, pp. 761-792, December, 2014.

[12] I. Garcia, Y.-K. Ng, Eliminating Redundant and Less-Informative Rss News Articles Based on Word Similarity and a Fuzzy Equivalence Relation, *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, Arlington, VA, 2006, pp. 465-473.

[13] Google, *Stop Words Project*, http://code.google.com/p/stop-words/.

[14] R. Hahn, C. Bizer, C. Sahnwaldt, C. Herta, S. Robinson, M. Bürgle, H. Düwiger, U. Scheel, Faceted Wikipedia Search, *Proceedings of the 13th International Conference on Business Information Systems*, Berlin, Germany, 2010, pp. 1-11.

[15] S. A. Hale, Multilinguals and Wikipedia Editing, *Proceedings of the 2014 ACM Conference on Web Science, Bloomington*, IN, 2014, pp. 99-108.

[16] P. Hazel, *Pcre-Perl Compatible Regular Expressions*, from http://www.pcre.org/pcre.txt.

[17] T. Hofmann, Probabilistic Latent Semantic Indexing, *Proceedings of the 22nd Annual International SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, 1999, pp. 50-57.

[18] T. Hofmann, Unsupervised Learning by Probabilistic Latent Semantic Analysis, *Machine Learning*, Vol. 42, No. 1, pp. 177-196, January, 2001.

[19] T. Hofmann, Collaborative Filtering Via Gaussian Probabilistic Latent Semantic Analysis, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, Toronto, Canada, 2003, pp. 259-266.

[20] T. Hofmann, B. Schölkopf, A. J. Smola, Kernel Methods in Machine Learning, *The Annals of Statistics*, Vol. 36, No. 3, pp. 1171-1220, 2008.

[21] J.-W. Hsieh, L.-C. Chen, S.-Y. Chen, D.-Y. Chen, S. Alghyaline, H.-F. Chiang, Vehicle Color Classification under Different Lighting Conditions through Color Correction, *IEEE Sensors Journal*, Vol. 15, No. 2, pp. 971-983, February, 2015.

[22] K. Ishida, T. Ohta, An Approach for Organizing Knowledge According to Terminology and Representing It Visually, *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, Vol. 32, No. 4, pp. 366-373, November, 2002.

[23] Z. Ji, P. Jing, J. Wang, Y. Su, Scene Image Classification with Biased Spatial Block and Plsa, *International Journal of Digital Content Technology and its Applications*, Vol. 6, No. 1, pp. 398-404, January, 2012.

[24] R. Klein, A. Kyrilov, M. Tokman, Automated Assessment of Short Free-Text Responses in Computer Science Using Latent Semantic Analysis, *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, Darmstadt, Germany, 2011, pp. 158-162.

[25] R. Krestel, P. Fankhauser, W. Nejdl, Latent Dirichlet

Allocation for Tag Recommendation, *Proceedings of the Third ACM Conference on Recommender Systems*, New York, 2009, pp. 61-68.

[26] T. K. Landauer, P. W. Foltz, D. Laham, An Introduction to Latent Semantic Analysis, *Discourse Processes*, Vol. 25, No. 2-3, pp. 259-284, 1998.

[27] T. K. Landauer, D. S. McNamara, S. Dennis, W. Kintsch, *Handbook of Latent Semantic Analysis*, Psychology Press, 2013.

[28] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, C. Bizer, Dbpedia: A Large-Scale, Multilingual Knowledge Base Extracted from Wikipedia, *Semantic Web Journal*, Vol. 6, No. 2, pp. 167-195, April, 2015.

[29] M. Li, W. K. Li, G. Li, On Mixture Memory Garch Models, *Journal of Time Series Analysis*, Vol. 34, No. 6, pp. 606-624, November, 2013.

[30] M. Liénou, H. Maître, M. Datcu, Semantic Annotation of Satellite Images Using Latent Dirichlet Allocation, *IEEE Geoscience and Remote Sensing Letters*, Vol. 7, No. 1, pp. 28-32, January, 2010.

[31] T. Li, S. Ma, M. Ogihara, Document Clustering Via Adaptive Subspace Iteration, *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, United Kingdom, 2004, pp. 218-225.

[32] M. Lintean, C. Moldovan, V. Rus, D. McNamara, The Role of Local and Global Weighting in Assessing the Semantic Similarity of Texts Using Latent Semantic Analysis, *Proceedings of the 23rd International Florida Artificial Intelligence Research Society Conference*, Daytona Beach, FL, 2010, pp. 235-240.

[33] Z. Liu, Y. Zhang, E. Y. Chang, M. Sun, Plda+: Parallel Latent Dirichlet Allocation with Data Placement and Pipeline Processing, *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, pp. 26:21-26:18, April, 2011.

[34] Y. Lu, Q. Mei, C. X. Zhai, Investigating Task Performance of Probabilistic Topic Models: An Empirical Study of Plsa and Lda, *Information Retrieval*, Vol. 14, No. 2, pp. 178-203, April, 2011.

[35] Z. Lu, Y. Peng, H. S. Ip, Image Categorization Via Robust Plsa, *Pattern Recognition Letters*, Vol. 31, No. 1, pp. 36-43, January, 2010.

[36] K. Makita, H. Suzuki, D. Koike, T. Utsuro, Y. Kawada, T. Fukuhara, Labeling Blog Posts with Wikipedia Entries through Lda-Based Topic Modeling of Wikipedia, *Journal of Internet Technology*, Vol. 14, No. 2, pp. 297-306, March, 2013.

[37] J. McInerney, A. Rogers, N. R. Jennings, Improving Location Prediction Services for New Users with Probabilistic Latent Semantic Analysis, *Proceedings of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh*, Pennsylvania, 2012, pp. 906-910.

[38] A. Mesaros, T. Heittola, A. Klapuri, Latent Semantic Analysis in Sound Event Detection, *Proceedings of the 19th European Signal Processing Conference*, Barcelona, Spain, 2011, pp. 1307-1311.

[39] D. Milne, I. H. Witten, An Open-Source Toolkit for Mining Wikipedia, *Artificial Intelligence*, Vol. 194, No. 1, pp. 222-239, January, 2013.

[40] M. Naughton, N. Stokes, J. Carthy, Sentence-Level Event Classification in Unstructured Texts, *Information Retrieval*, Vol. 13, No. 2, pp. 132-156, April, 2010.

[41] H. V. Nguyen, L. Bai, Cosine Similarity Metric Learning for Face Verification, *Lecture Notes in Computer Science*, Vol. 6493, pp. 709-720, 2011.

[42] M. G. Ozsoy, F. N. Alpaslan, I. Cicekli, Text Summarization Using Latent Semantic Analysis, *Journal of Information Science*, Vol. 37, No. 4, pp. 405-417, August, 2011.

[43] M. F. Porter, *Snowball: A Language for Stemming Algorithms*, http://snowball.tartarus.org/texts/introduction.html.

[44] B. Shen, Y.-S. Zhao, An Experimental Study of Incremental Svd on Latent Semantic Analysis, *Journal of Internet Technology*, Vol. 15, No. 1, pp. 35-41, January, 2014.

[45] Y. Shen, X. He, J. Gao, L. Deng, G. Mesnil, A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, Shanghai, China, 2014, pp. 101-110.

[46] A. Siddiqui, N. Mishra, J. S. Verma, A Survey on Automatic Image Annotation and Retrieval, *International Journal of Computer Applications*, Vol. 118, No. 20, pp. 27-32, May, 2015.

[47] K. Somasundaram, G. C. Murphy, Automatic Categorization of Bug Reports Using Latent Dirichlet Allocation, *Proceedings of the fifth India Software Engineering Conference*, Kanpur, India, 2012, pp. 125-130.

[48] J. Speh, A. Muhic, J. Rupnik, Parameter Estimation for the Latent Dirichlet Allocation, *Proceedings of the 2013 Conference on Data Mining and Data Warehouses*, Ljubljana, Slovenia, 2013, pp. 1-4.

[49] V. K. R. Sridhar, Unsupervised Topic Modeling for Short Texts Using Distributed Representations of Words, *Proceedings of the First Workshop on Vector Space Modeling for Natural Language Processing*, Denver, Colorado, 2015, pp. 192-200.

[50] P.-N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Boston, Massachusetts, Addison-Wesley Press, 2005.

[51] C. Wang, D. M. Blei, Variational Inference in Nonconjugate Models, *Journal of Machine Learning Research*, Vol. 14, No. 1, pp. 1005-1031, January, 2013.

[52] Wikipedia, *Wikipedia: What Wikipedia Is Not*, http://tinyurl.com/qbdajlz.

[53] F. Wu, D. S. Weld, Open Information Extraction Using Wikipedia, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 2010, pp. 118-127.

[54] C. Xu, Y.-F. Zhang, G. Zhu, Y. Rui, H. Lu, Q. Huang, Using Webcast Text for Semantic Event Detection in Broadcast Sports Video, *IEEE Transactions on Multimedia*, Vol. 10, No. 7, pp. 1342-1355, November, 2008.

[55] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, I-H. Meng, Text Summarization Using a Trainable Summarizer and Latent Semantic Analysis, *Information Processing and Management*, Vol. 41, No. 1, pp. 75-95, January, 2005.

Biography

Lin-Chih Chen is an associate professor in the Department of Information Management at National Dong Hwa University, Taiwan. His research interests include Web Intelligent and Web Technology. He develops many Web Intelligent systems include Cayley search engine, On-The-Fly Document Clustering, LI keyword suggestion system, WSC clustering system, Cayley Scholar.