Unsupervised Contextual Keyword Relevance Learning and Measurement using PLSA

Sudarsun. S, *Member, IEEE*, Dalou Kalaivendhan, Venkateswarlu. M

Abstract-- **In this paper, we have developed a probabilistic approach using PLSA for the discovery and analysis of contextual keyword relevance based on the distribution of keywords across a training text corpus. Since the strength of these relationships is measured in terms of probabilities, we are able to use probabilistic inference to perform a variety of analyses, including such tasks as adaptive document segmentation, keywords classification, and collaborative recommendation. We have also shown experimentally, the flexibility of this approach in classifying keywords into different domains based on their context. We will explore these issues through the construction of PLSA models for huge corpora and by deploying techniques for overcoming expected bottlenecks. We have developed a prototype system that allows us to project keyword queries on the loaded PLSA model and returns keywords that are closely correlated. The keyword query is vectorized using the PLSA model in the reduce aspect space and correlation is derived by calculating a dot product. We also discuss the parameters that control PLSA performance including a) number of aspects, b) number of EM iterations c) weighting functions on TDM (pre-weighting) and their role in the quality of relevancy estimates. We have estimated the quality through computation of precision-recall scores. We have performed various experiments on PLSA models built both on varying corpus sizes and varying document domains count. Finally, we present a step-by-step procedure to build and tune a de novo PLSA model.**

Index Terms-- **SVD, Synonymy, Polysemy, Unsupervised Clustering, PLSA, Aspect model, Keyword Relevance**

I. Introduction

The primary goal of Contextual Keyword Relevance Learning is to leverage information surrounding a query to identify keywords related to the specific search terms. Standard approaches, such as SVD, document clustering and association rules, do not generally provide the ability to automatically characterize or quantify those unobservable factors that can often drive common patterns. Although SVD provides an unsupervised solution for building associations between keywords by addressing problem of synonymy, it suffers from a different set of problems related to polysemy within text data. Probabilistic Latent Semantic Analysis (PLSA) is particularly useful in this context, since it can uncover non-obvious associations among keywords based on the contextual co-occurrence patterns of these keywords in documents. In general, any text can be seen as a distribution of words bound by some context or topic(s), which could be either explicit or implicit. PLSA is designed to uncover hidden topics, which it assumes to be responsible for the word distribution found in a text. That is, PLSA is based on the promise that a document text is composed of topics which, in turn, drive word distribution (aspect model). If we consider the reverse, words falling into the same topic should be considered more relevant than words falling in different topics.

Information retrieval [9], [10] is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information. It deals with extraction of unstructured data, especially from textual documents, in response to an often-unstructured query or topic statement. One such application of information retrieval is in the task of word association [6]. When some word is unclear, people can clarify themselves by selecting from among the related words of its category. For example, in computer-related querying, the word '*network*' is the most closely related response by our system for the word *'Novell*'. Although these words are not synonyms, they fall in the same category according to our classification. But hidden connections between words as well as issues of domain specificity and polysemy, among many others, all conspire to confound classification

Overcoming these obstacles requires the development of techniques for automatically classifying keywords with the same contextual meaning in their correct domain as well as to discover hidden semantic relationships among keywords and between keywords and documents. A general method for capturing the latent or hidden semantic associations among co-occurring objects is Latent Semantic Analysis (LSA) [12], [13]. This method is mostly used in automatic indexing and information retrieval, where LSA usually takes the high dimensional vector space representation of documents and applies a dimension reducing linear projection, such as Singular Value Decomposition (SVD) [12] to generate a reduced latent space representation. Probabilistic Latent Semantic Analysis (PLSA) [1]-[3] models, proposed by Hofmann, provide a probabilistic approach for the discovery of latent variables that is more flexible and has a more solid statistical foundation than the standard LSA. The basis of PLSA is a model often referred to as the *aspect model* [11]. Assuming that there exist a set of hidden factors underlying the co-occurrences between two sets of objects, PLSA uses an Expectation-Maximization (EM) algorithm [11] to estimate the probability values measuring the relationships between the hidden factors and the two sets of objects. Due to its great flexibility, PLSA has been widely and successfully used across a variety of application domains, including information retrieval, text learning, and co-citation analysis.

In this paper, we propose an approach to find the related keywords for a given query based on the PLSA model using the contextual co-occurrence exists between the keywords and

This work was supported by Checktronix India Pvt Ltd, Chennai.

documents. In this paper, we refer to the hidden factors that represent the latent relationships among these entities as aspects [11]. By applying the PLSA model, we can effectively identify and classify these hidden factors, thus quantitatively measuring the relationships between keywords and aspects, as well as between documents and aspects. Since these relationships are measured in terms of probabilities, we could use probabilistic inference to perform analysis tasks such as document segmentation [4],[5] & keywords classification [6].

A. Organization of Paper

The paper is organized as follows. In Section 2 we lay out the different existing applications and other works related to or based on the PLSA Method. In Section 3 we provide a technical overview of the Probabilistic Latent Semantic Analysis model as applied to unsupervised contextual keyword relevance learning. In Section 4 we present the details of the word association method for retrieving the related terms for a given query term and using them to classify external documents into their respective domains. Section 5 gives the details of the different pre-weighting algorithms applied on the Document-Term matrix. Finally, we present our experiments and interpretation of the result in Section 6 and conclude the paper in Section 7. The Appendix provides the step-by-step procedure for building a PLSA Model as well as details of the issues involved in and the solution for handling huge frequency matrices and for control of iterations in EM Algorithm.

II. RELATED WORK

Hofmann [1]-[3] explains the basics of the PLSA method as applied within different solutions. In our work we have used this method for our application according to our needs. Thomas Hofmann explained the usage of the PLSA Method [4], [5] for finding the similarities between documents. Our paper explains the classification of external documents into respective domains by using the scalar product method. Work by Xin Jin et al. [7] describes the application of Probabilistic Latent Semantic Indexing (PLSI) method for web data usage mining. However, their focus is on user segment identification – that is, identifying the underlying tasks of a user session and integration of usage patterns with Web content information. Our work, on the other hand, involves deriving Contextual Keyword Relevance by using the word association method [6] for a given query terms in their context. Other related works include those of Tuomo Kakkonen et al. [8], whose approach tends to be of a more theoretical nature than this work. More specifically, their work is to find the grade of external document depends on the trained documents.

III. OVERVIEW OF PLSA METHOD

The overall process of contextual keyword relevance learning consists of three phases: 1) Building a Term-Document Matrix of the target text corpus; 2) Computing the aspect model parameters; 3) Keyword query projection and estimation or relevancy. The usage data-preprocessing phase results in a set of ***m*** *documents*, $D = \{ d_1, d_2, \ldots, d_m \}$ and a set of ***n*** *keywords*, $W = \{w_1, w_2, \ldots, w_n\}$. The document can be conceptually viewed as an ***mxn*** *Document-Term* matrix $DW=|n(d_i, w_j)|_{mxn}$, where $n(d_i, w_j)$ represents the weight of keyword w_j in a document d_i. The weights represent the existence or non-existence of the keyword in the document, or they may be a function of the occurrence or duration of the keyword in that document. The core of PLSA is a latent variable model [6], [10], otherwise called as an aspect model which associates hidden (unobserved) factor variable $Z=\{z_1,z_{2,\ldots,}z_l\}$ with observations in the co-occurrences data. In our context, each observation corresponds to keyword in a particular document.

The probabilistic latent factor model can be described as the following generative model; 1) Select a document d_i from D with probability $Pr(d_i)$, 2) Pick a latent factor z_k with probability $Pr(z_k|d_i)$, 3) Generate a word w_j from W with probability $Pr(w_j|z_k)$. As a result we obtain an observed pair (d_i,w_j), while the latent factor variable z_k is discarded. Translating this process into a joint probability model results in the following:

$$\Pr(d_i, w_j) = \Pr(d_i) * \Pr(w_j \mid d_i), \quad (1)$$

where,

$$\Pr(w_j \mid d_i) = \sum_{k=1}^{l} \Pr(w_j \mid z_k) * \Pr(z_k \mid d_i) \quad (2)$$

Summing over all possible choices of z_k from which the observation could have been generated. Using Bayes rule, it is straightforward to transform the joint probability into

$$\Pr(d_i, w_j) = \sum_{k=1}^{l} \Pr(z_k) * \Pr(d_i \mid z_k) * \Pr(w_j \mid z_k) \quad (3)$$

Now, in order to explain a set of observations (D, W), we need to estimate the parameters $Pr(z_k)$, $Pr(d_i|z_k)$ and $Pr(w_j|z_k)$, while maximizing the following likelihood $L(D,W)$ of the observations,

$$L(D,W) = \sum_{i=1}^{m} \sum_{j=1}^{n} n(d_i, w_j) * \log \Pr(d_i, w_j) \quad (4)$$

Expectation-Maximization (EM) algorithm is a well-known approach to performing maximum likelihood parameter estimation in latent variable models. It alternates two steps: (1) an expectation (E) step where posterior probabilities are computed for latent variables, based on the current estimates of the parameters, (2) a maximization (M) step, re-estimate the parameters in order to maximize the expectation of the complete data likelihood. The EM algorithm begins with some initial values of $Pr(z_k)$, $Pr(d_i|z_k)$, and $Pr(w_j|z_k)$. In the E-Step we compute:

$$\Pr(z_k \mid d_i, w_j) = \frac{\Pr(z_k) * \Pr(d_i \mid z_k) * \Pr(w_j \mid z_k)}{\sum_{k=1}^{l} \Pr(z_k) * \Pr(d_i \mid z_k) * \Pr(w_j \mid z_k)} \quad (5)$$

Or, can be computed as

$$\Pr(z_k \mid d_i, w_j) = \frac{\Pr(w_j \mid z_k) * \Pr(z_k \mid d_i)}{\sum_{k=1}^{l} \Pr(w_j \mid z_k) * \Pr(z_k \mid d_i)} \quad (6)$$

By applying Tempered EM (TEM) algorithm [10] the posterior probability can be calculated as follows

$$\Pr_\beta(z_k \mid d_i, w_j) = \frac{\Pr(z_k) * [\Pr(d_i \mid z_k) * \Pr(w_j \mid z_k)]^\beta}{\sum_{k^1=1}^{l} \Pr(z_{k^1}) * [\Pr(d_i \mid z_{k^1}) * \Pr(w_j \mid z_{k^1})]^\beta} \quad (7)$$

Or

$$\Pr_\beta(z_k \mid d_i, w_j) = \frac{[\Pr(w_j \mid z_k) * \Pr(z_k \mid d_i)]^\beta}{\sum_{k^1=1}^{l} [\Pr(w_j \mid z_{k^1}) * \Pr(z_{k^1} \mid d_i)]^\beta} \quad (8)$$

where β is controlling parameter (inverse computational temperature). In the maximization step, we aim at maximizing the expectation of the complete data likelihood $E(L^C)$,

$$E(L^C) = \sum_{i=1}^{m}\sum_{j=1}^{n} n(d_i, w_j) * \sum_{k=1}^{l} \Pr(z_k \mid d_i, w_j) \log \Pr(d_i, w_j) \quad (9)$$

While taking into account the constraints, $\sum_{k=1}^{l} \Pr(z_k) = 1$, on the factor probabilities, as well as the following constraints on the conditional probabilities:

$$\sum_{k=1}^{l} (\sum_{i=1}^{m} \Pr(d_i \mid z_k) - 1) = 0, \quad (10)$$

$$\sum_{k=1}^{l} (\sum_{i=1}^{m} \Pr(z_k \mid d_i) - 1) = 0, \quad (11)$$

And

$$\sum_{k=1}^{l} (\sum_{j=1}^{n} \Pr(w_j \mid z_k) - 1) = 0. \quad (12)$$

Through the use of Lagrange multipliers, we can solve the constraint maximization problem to get the following equations for re-estimated parameters:

$$\Pr(z_k) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} n(d_i, w_j) * \Pr(z_k \mid d_i, w_j)}{\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k^1=1}^{l} n(d_i, w_j) * \Pr(z_{k^1} \mid d_j, w_j)} \quad (13)$$

Or

$$\Pr(z_k) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} n(d_i, w_j) * \Pr(z_k \mid d_i, w_j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} n(d_i, w_j)} \quad (14)$$

$$\Pr(w_j \mid z_k) = \frac{\sum_{i=1}^{m} n(d_i, w_j) * \Pr(z_k \mid d_i, w_j)}{\sum_{i=1}^{m}\sum_{j^1=1}^{n} n(d_i, w_{j^1}) * \Pr(z_k \mid d_i, w_{j^1})} \quad (15)$$

$$\Pr(z_k \mid d_i) = \frac{\sum_{j=1}^{n} n(d_i, w_j) * \Pr(z_k \mid d_i, w_j)}{\sum_{i^1=1}^{m}\sum_{j=1}^{n} n(d_{i^1}, w_j) * \Pr(z_k \mid d_{i^1}, w_j)} \quad (16)$$

Iterating the above computation of expectation and maximization steps monotonically increases the total likelihood of the observed data $L(D, W)$ until an optimal solution is reached. The computational complexity of this algorithm is $O(mnl)$, where m is the number of documents, n is the number of key terms, and l is the number of aspects.

Since the usage observation matrix (Document-Term Matrix) is, in general, very sparse, the memory requirements can be dramatically reduced using efficient sparse matrix representation of the data. So we represented this matrix in Harwell-Boeing (HB) sparse matrix format and Inverted Index format for computational efficiency purpose. It has been observed the latter sparse matrix format is taking less memory and less computational time for building the PLSA model.

IV. DISCOVER AND ANALYSIS OF USAGE PATTERNS WITH PLSA METHOD

This section explains about probabilistic inference to perform a variety of analysis tasks such as adaptive document segmentation, keywords classification, as well as predictive tasks such as collaborative recommendations.

A. Query Projection

Using word association method does query projection. In this method we calculate the word association (conditional) probability between the given query term and terms appearing in the vocabulary to represent the documents. The main constraint in query projection is the query term must exist in vocabulary.

$$\Pr(w_j \mid w_{query}) = \sum_{k=1}^{l} \Pr(w_j \mid z_k) * \Pr(z_k \mid w_{query}) \quad (17)$$

where l is the number of aspects and w_{query} is the query term index from vocabulary. The equation involves multiplication of $\Pr(w_j|z_k)$ and $\Pr(z_k|w_{query})$. $\Pr(w_j|z_k)$ is already been calculated by using PLSA algorithm and $\Pr(z_k|w_{query})$ can be calculated by using Bayes rule as,

$$\Pr(z_k \mid w_{query}) = \frac{\Pr(w_{query} \mid z_k) * \Pr(z_k)}{\Pr(w_{query})} \quad (18)$$

Since the value of $\Pr(w_{query})$ will remain constant throughout the calculation, the equation can be rewritten as,

$$\Pr(z_k \mid w_{query}) \approx \Pr(w_{query} \mid z_k) * \Pr(z_k) \quad (19)$$

B. Classification of Test Document using Ideal Documents

In this method we prepare the ideal document for every domain manually and it is used to get the ideal vector V_I for every domain using the PLSA method. It generates the document vector for test document V_T to be classified using PLSA method. The similarities between the domain in the

model and a test document folded in to the model can be calculated with the scalar product between ideal vector V_I for every domain and vector for test document V_T . The domain, which gets the maximum score, can be treated as the domain of the test document.

C. Classification of Test Document using Centroid Method

This method involves building of separate PLSA models for all our training domains. We generated PLSA models for individual domains to get $Pr(z_k|d_i)$ distribution per domain. The ideal vector $Pr(z_k|ideal)$ can be gotten from the centroid calculated by finding the mean of all hidden aspects of all documents in a domain. For a test document, computing the scalar product of $Pr(z_k|q)$ and $Pr(z_k|ideal)$ gets a score per domain. We may classify the test document to the domain that had got the higher score [8], [9].

D. Classification of Test Document based on Hidden Aspects

This method fully depends of the aspect values generated for a test document projected on the PLSA model. We built a global PLSA model using documents from all our training domains. Using the PLSA models keyword relevancy measure, we had tuned the model by correcting the choice of hidden aspects, number of iterations and weighting functions. We build a BPN based neural network, which had an input nodes count matches the number of aspects and output node count matching the number of target domains. We used the training corpus used of PLSA model building as the training corpus for neural network as well. The steps involves; 1) Project the training document on the PLSA model. 2) Use the vector (of aspects) generated by PLSA model as input to the neural net for training. We repeated the above steps for a subset of the training corpus of PLSA model covering all the domains to be learned by neural net. By this process, the BPN learns about the classification based on the vectors generated by the PLSA model, which was found to improve the classification to a greater extent. While testing, 1) a test document is fed to PLSA for projection and vectorization. 2) the vector (of aspects) is fed to the BPN neural network. 3) Based on which output node of BPN fire, we classified the document to the appropriate domain.

V. WEIGHTING ALGORITHMS

We tried various weighting algorithms and compared their performances by varying the other parameters. We are presenting the weighting functions we had used.

A. Normal Weighting

Normal weighting scales down the column vector so that column sums up to one in order to nullify the effect of extreme values.

$$n(d_i, w_j) = \frac{n(d_i, w_j)}{\sum_{i=1}^{m} n(d_i, w_j)} \tag{21}$$

where $n(d_i, w_j)$ → Frequency of term 'j' in document 'i'.

B. Normalized Document Vector Weighting

This is a LWF, which makes the column vector (document) of unit length.

$$n(d_i, w_j) = \frac{n(d_i, w_j)}{\sqrt{\sum_{i=1}^{m} (n(d_i, w_j))^2}} \tag{22}$$

C. Weighted Inverse Document Frequency Weighting

It is a LWF that normalizes the term vector based on sum of the frequencies of that particular term vector. Normal weighting scales down the row vector so that row sums up to one in order to nullify the effect of extreme values.

$$n(d_i, w_j) = \frac{n(d_i, w_j)}{\sum_{j=1}^{n} n(d_i, w_j)} \tag{23}$$

D. Inverse Word Frequency Weighting

IWF is similar to IDF but it calculates weight based on Frequency of particular word across the documents.

$$n(d_i, w_j) = n(d_i, w_j) * \log \left[\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} n(d_i, w_j)}{\sum_{j=1}^{n} n(d_i, w_j)} \right] \tag{24}$$

E. Inverse Document Weighting

IDF is a GWF, which suppresses the importance of words by allocating less weight to words that often occur and boosts the weight of words that occur rarely in the entire corpus.

$$n(d_i, w_j) = n(d_i, w_j) * \log \frac{N}{FN(d_i)} \tag{25}$$

where N → Total Number of Documents in the corpus

$FN(d_i)$ → Number of words in i^{th} document.

E Composite Weighting Function

Composite Weighting Function is, applying both LWF and GWF on the Document-Term matrix. For this experiment, we tried two combination of this weighting function (i.e.) IWF+NDV and IDF+NDV.

VI. EXPERIMENTS WITH PLSA MODEL

In this section, we discuss the various experiments we performed on PLSA Method. We used different sets of corpus to perform experiments with our PLSA-based Contextual Keyword Relevance Learning and external document classification into its respective domain. We have collected 100,000 documents from 20 different domains, nearly 5,000 documents per each domain. Documents for all the domains are equally populated in the training set. Most of the documents for the rare domains are collected from the web and reviewed manually and then populated in the training corpus. We have applied above-mentioned weighting algorithms on raw Document-Term matrix and built the PLSA Model. This process has been repeated for different iterations, aspects and beta values.

The efficiency of the system to classify keywords in their correct domains is measured by calculating precision and recall values of the system.

Precision is the fraction of the search output that is relevant for a particular query. Thus precision can be calculated as:

precision = relevant records retrieved / total records (26)

In terms of confusion matrix notation,

$$precision = \frac{TP}{(TP + FP)} \quad (27)$$

The recall on the other hand is the ability of a retrieval system to obtain all or most of the relevant documents in the collection. Thus it requires knowledge not just of the relevant and retrieved but also those not retrieved

Recall = relevant retrieved / total relevant records (28)

In terms of confusion matrix notation,

$$recall = \frac{TP}{(TP + FN)} \quad (29)$$

A. Effect of Corpus Size

The corpus size is varied from the range of 10,000 to 100,000 by keeping aspects and iterations as common. Varying the weighting algorithms, various experiments have been performed and they are plotted as shown below.

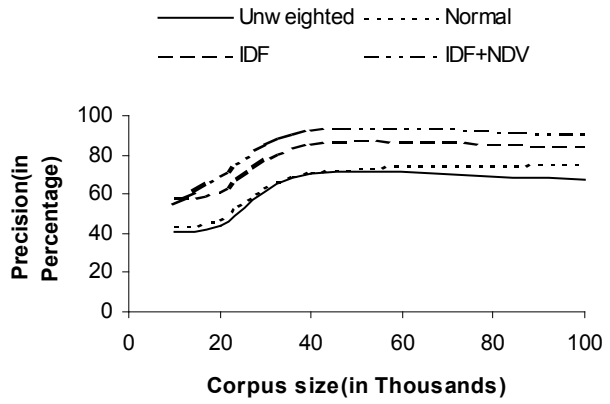

Fig. 1. Precision Vs Corpus size

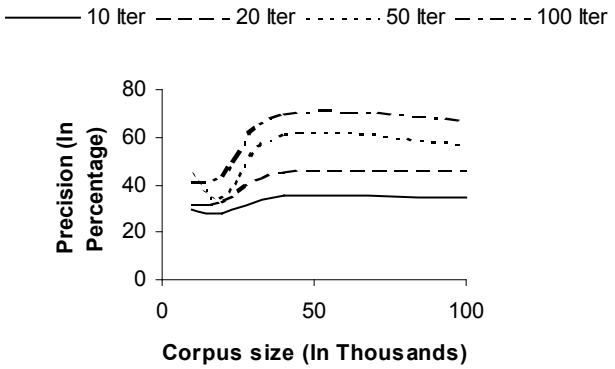

Fig. 2. Precision Vs Corpus size

This above graph illustrates the effect on precision value with the increase in size of corpus for various EM iterations.

B. Effect of number of Aspects

The number of aspects has been varied from the range of 50 to 300 by keeping domain and iterations constant. Differing the weighting algorithms, the experiments have been performed and they are plotted as shown below.

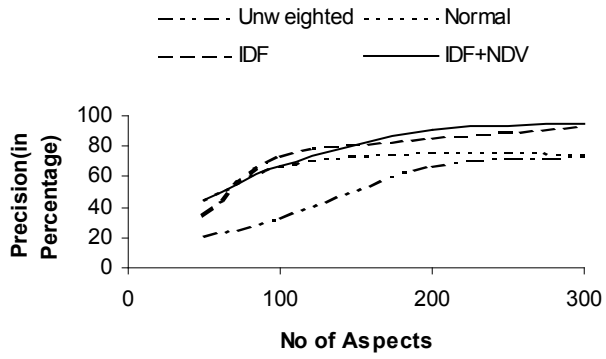

Fig. 3. Precision Vs. Number of aspects

From the graph, it is evident that the weight algorithms yield the greatest impact in producing the accurate result. The IDF+NDV weighting algorithm produces better results. For 100,000 document corpora, maximum precision was achieved when aspects were set to 300

C. Effect of number of Iterations

Similarly the number of iterations has been varied from the range 10 to 100 by keeping the aspects count constant. Differing the size of corpus, the experiments have been carried out and shown below.

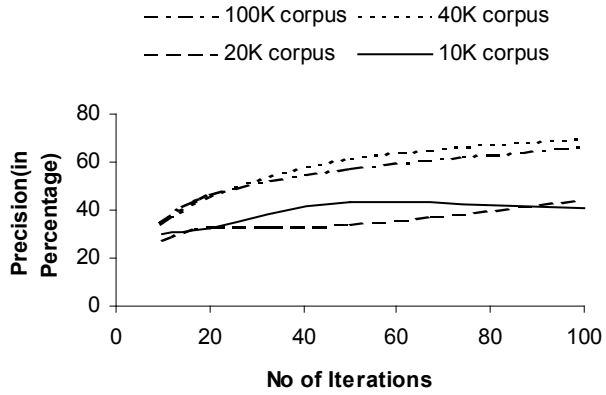

Fig. 4. Precision Vs Number of Iterations.

This graph produces good result for 40,000 corpus because of the chosen aspect size being 200. 100,000 documents corpus will produce good result than 40,000 documents corpus if the aspect size is set to 300. But when the corpus size is small, the aspects should be chosen judiciously.

D.Result of Classification of Test Documents

We have tested our system by varying corpus size from 1,000 to 5,000 for 5 domains and tested the document classification for every domain. The results are plotted in the following graphs.

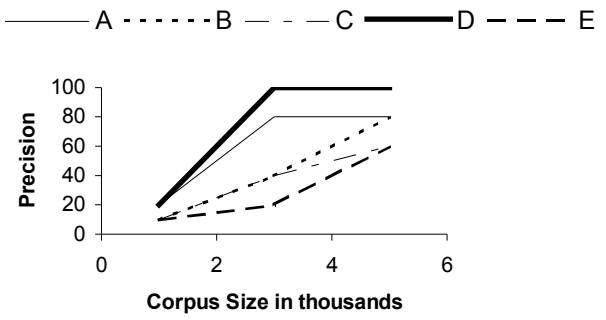

Fig. 5. Document Classification by using Centroid Method
A=Accounts, B=Administration, C=Call Center D=IT And E=Hospitality

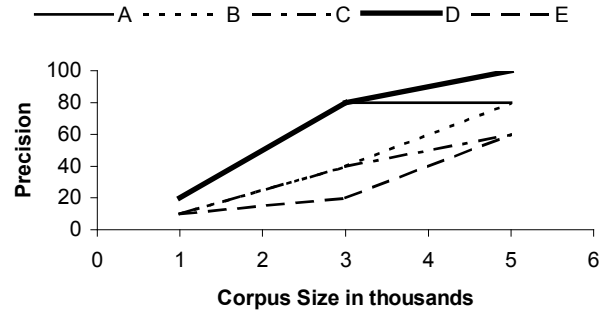

Fig. 6. Document Classification by using Ideal Document Method

VII. CONCLUSION

In this paper, we have presented a probabilistic approach using PLSA for the discovery and analysis of contextual keyword relevance based on a words distribution extracted from a training text corpus. We have shown the flexibility of this approach in classifying keywords in their correct domains. We had also presented the results of various experiments performed on the parameters that control PLSA performance viz. a) Aspects, b) EM Iterations c) Weighting functions (Pre-Weighting). We have discussed the methods for building and deploying PLSA-based models and then demonstrated their results.

VIII. APPENDIX

A. Step-by-Step Procedure to Build PLSA Model

The procedure is as follows: 1) Build TDM of the target corpus; 2) Convert the matrix to HB or Inverted Index format; 3) Apply a suitable weighting function; 4) Compute the aspects model parameters by analyzing EM iterations; 5) Log-Likelihood may be used as the convergence condition.

B. Issues and Solutions for Handling Huge Frequency Matrix

TDM is very sparse in nature (<1% dense). It is wise to represent the matrix in HB format or the Inverted Index format adopted by Monay [14]. The former gives best memory image but takes significantly more time than the later.

C. Controlling number of EM iterations

The convergence condition is generally dictated by the Log-Likelihood estimate. In equation (4) we calculated $\log(\Pr(d_i,w_j))$ which yielded correct results only while comparing documents and keywords. We adopted manual control on no of iterations, as the method does not perform well for keyword-keyword comparison.

IX. REFERENCES

[1] Thomas Hofmann, "Probabilistic Latent Semantic Indexing," Proc. 22 Int'l SIGIR Conf. on Research and Development in Information Retrieval, 1999.

[2] Thomas Hofmann, "Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning," pp.47 (1): 177–196,2001.

[3] Thomas Hofmann, "Probabilistic Latent Semantic Analysis," Proc. Uncertainty in Artificial Intelligence, 1999.

[4] Thomas Hofmann, "Learning the Similarity of Documents: An Information-Geometric Approach to Document Retrieval and categorization," pp.914-920, 2000.

[5] T. Hofmann and J. Puzicha, "Unsupervised learning from dyadic data. Technical report," 1998.

[6] Scott Brown, Mark Steyvers, *Probabilistic Topic Models*. Available: http://oz.ss.uci.edu/237/hwk/topics.html

[7] Xin Jin, Yanzan Zhou, and Bamshad Mobasher, "Web Usage Mining Based on Probabilistic Latent Semantic Analysis"

[8] Tuomo Kakkonen, Niko Myller, Jari Timonen, and Erkki Sutinen, "Automatic Essay Grading with Probabilistic Latent Semantic Analysis".

[9] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan, "Web usage mining: Discovery and applications of usage patterns from web data", proc. SIGKDD Explorations, pp.1 (2):12− 23, 2000.

[10] R. Cooley, B. Mobasher, and J. Srivastava, "Web mining: Information and pattern discovery on the World Wide Web."

[11] Dempster, A., Laird, N., and Rubin, D. Maximum, "likelihood from incomplete data via the EM algorithm," pp.1-38, 1977.

[12] Dumais, S. T, "Latent semantic indexing (LSI)," Proc. The Text Retrieval Conf. TREC-3, pp. 219-30, 1995.

[13] Michael W. Berry, Susan T. Dumais and Todd A.Letsche, Computational "Methods for Intelligent Information Access".

[14] Florant Monay, *PLSA C Code.* Available: http://www.idiap.com/~monay

X. BIOGRAPHIES

Sudarsun S (M' 2002) is the Director – R & D at Checktronix India Pvt Ltd, Chennai. He holds an M.Tech Computer Science from IIT Madras. He received a BE degree in Electronics and Instrumentation Engineering from Madras University with a Gold Medal. His research experience includes Statistical Natural Language Processing, Machine Learning and Distributed Computing.

Dalou Kalaivendhan received his B.Tech in Computer Science and Engineering as well as his M.Tech in Distributed Computing Systems from Pondicherry University. He is currently a Research Associate in Checktronix India Pvt Ltd. His research interests include Algorithms, Information Retrieval and Object Oriented Quality Prediction.

Venkateswarlu Malapati is a Research Associate at Checktronix India Pvt Ltd, Chennai. He completed his B.Tech in Information Technology from Pondicherry University in 2005. His research interests include Data Mining, PLSA and Pattern Classification.