International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 24 (2015) pp 43942-43946

An Efficient Query Retrieval Model Using Pattern Tree-Based Latent Dirichelt Allocation

Varghese . S .Chooralil

Research Scholar, Department of Computer Science and Engineering, VELS University, Chennai-600043
Varghesee.kutty@gmail.com

E. Gopinathan

Dean, School of Engineering Department of Computer Science and Engineering VELS University ,Chennai-600043
dean.se@velsuniv.org

Abstract
Today with the advancement in Internet diversified retrieval is becoming very vast and user want to retrieve the exact information from many e-commerce sites, e.g. Flipkart, Snapdeal and so on. Certain number of times, the queries issued by the user and returned by the search engine is not entirely similar. Many prior works has been conducted in the area of information retrieval, however, optimization and time to retrieve the query still remains a significant challenge to be addressed. In this work, an efficient Pattern Tree-based Latent Dirichlet Allocation (PT-LDA) is designed with the objective of reducing the query retrieval time and therefore improving the query retrieval ratio. The Topic-based Pattern Tree in PT-LDA initially constructs a topic based pattern model using a relevance factor aiming at reducing the query retrieval time. Next, Probability-based Latent Dirichlet Allocation performs an efficient mapping using probability measure aiming at improving the query retrieval ratio. The Pattern Tree-based LDA algorithm in PT-LDA framework efficiently maps the association between user queries and web related documents in a significant manner. The efficiency of PT-LDA framework is measured using the Reuters-21578 Text Categorization Collection Data Set from UCI repository and is shown to be significantly more efficient in terms of query retrieval time, query retrieval ratio and accuracy of results being obtained when compared to the state-of-the-art works.

Keywords: Pattern Tree, Latent Dirichlet Allocation, Topic-based, Probability-based, Relevance factor, Probability measure

Introduction

The emergence of web query optimization and its extensions into the Internet has resulted in the significant impact on query retrieval rate. Many of the researchers have contributed towards web query optimization. Semantic Relation using Automatic Enrichment (SR-AE) [1] used Word Net's glossaries for efficient construction of knowledge base system with the aid of axioms. In [2], Automatic Discovery of Personal Name (ADPN) extracted larger amount of information with the aid of mining algorithms, but was not effective in higher order associations.

One of the key areas in Web is the Web search. Semantic processing [3] was described using semantic web search and efficiently evaluated simple and complex Web search queries in the web search engine. Different behavior patterns were extracted in an efficient manner using ontology language based on web [6] and efficiently explored different behavior patterns at minimum time.

Information Retrieval through keyword search model [4] was performed using sub graph that resulted in improved retrieval rate, but compromising the optimization factor. Expectation Maximization [5] was applied for efficient optimization from new unseen sites. With the increase in the success rate of information retrieval (IR) through web, keyword search has received greater attention. Statistical information were used in [9] based on XML data and addressed the issues related to keyword ambiguity.

With the increasing demands from the user and with the advantage of high accessibility dynamic web pages are receiving more advantage than the static web pages called as the deep web. An unsupervised, record matching method [7] was designed to identify duplicate data from multiple Web databases, resulting in high precision rate. Efficient construction of semantic queries [10] for specific domain was designed in relatively lesser amount of time obtained accuracy.

Due to the unpredictable nature of data, adaptive query processing is the key for efficient retrieval. A read join method [8] was introduced designed for accuracy maximization. With the significant popularity and increased growth of the Web, keyword-based search has become the most popular search in the search engines. An effective ranking and searching method [11] based on top-k queries were designed with the aid of semantic relationships. Though search space was reduced and accuracy but at the cost of complexity. To address issues related to complexity, a method based on optimization called MapReduce [12] was introduced which not only minimized the processing time but also proved to be scalable.

Based on the aforementioned methods and techniques, in this work, we introduce, an efficient Pattern Tree-based Latent Dirichlet Allocation (PT-LDA) for processing well designed and complex user queries and efficient retrieval with less query retrieval time. The rest of the paper is organized as follows. Section 2 provides an overview on how to use PT-

LDA framework by designing a framework with the design of Pattern Tree-based LDA algorithm. In Section 4, experimental settings are presented followed by Section 5 that includes the results and discussions of experiments with the help of table and graph. Finally, the paper concludes with concluding remarks in Section 6.

Design of Pattern Tree-based Latent Dirichlet Allocation

In this work a probabilistic model called Pattern Tree-based Latent Dirichlet Allocation is designed to extract the topics from a collection of documents. The objective behind the use of Pattern Tree-based Latent Dirichlet Allocation (PT-LDA) is that the document includes several topics with the probability of occurrence of topic with its own probability with respect to overall words in the document. Figure 1 shows the block diagram of Pattern Tree-based Latent Dirichlet Allocation.

As shown in the figure, the Pattern Tree-based Latent Dirichlet Allocation performs two steps. The first step Topic-based Pattern Tree is designed with the aim of reducing the query retrieval time using topic-based corpus and using relevance factor. The second step Probability-based Latent Dirichlet Allocation is designed aiming at improving the query retrieval ratio using probability measure. The elaborate description of PT-LDA framework is described in the forthcoming sections.

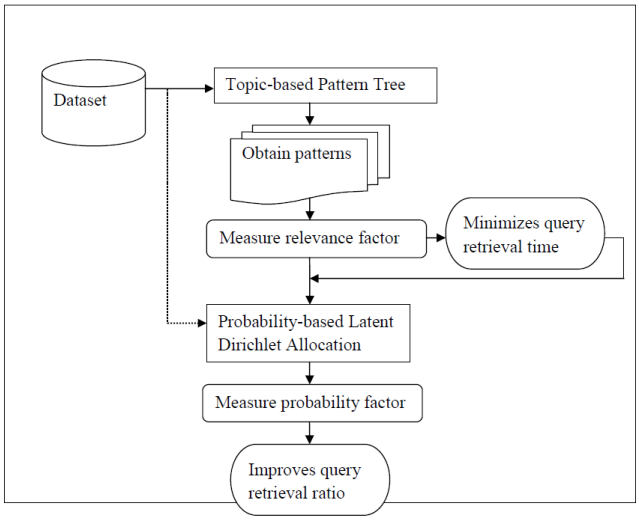

Figure 1 Block diagram of Pattern Tree-based Latent Dirichlet Allocation

Design of Topic-based Pattern Tree

The first step involved in the design of PT-LDA framework is the construction of Topic-based Pattern Tree. The Topic-based Pattern Tree is a model structured on the basis of topics or domain. Here the pattern tree is constructed according to the domain structure.

Let us consider a document 'Doc' which consists of several topics '$T_i = T_1, T_2, \ldots, T_n$' with overall words '$w_i = w_1, w_2, .., w_n$' respectively and user request queries '$Query_i = Query_1, Query_2, \ldots, Query_n$'. In order to retrieve the information requested by the user using PT-LDA, a corpus for that query is generated using the Pattern Tree model.

$$Query_i = \sum_{i=1}^{n} Doc\ (T_i w_i) \quad (1)$$

From (1), the query '$Query_i$' includes the topics from the document 'Doc' which consists of many words 'w_i'. The Topic-based pattern tree 'T' includes a doublet '$T = (p, q)$', where '$p = (V, E, r)$' is a rooted tree, and '$q = q_n, where\ q_n \in V$'. Here '$V$' form the vertices that contain the document 'Doc', 'E' the edges that contain topics 'T_i' with a root 'r' with a user query '$Query_i$'. Figure 2 shows Topic-based pattern tree representation, related to employee details of a company, where the topic form the root node'r'.

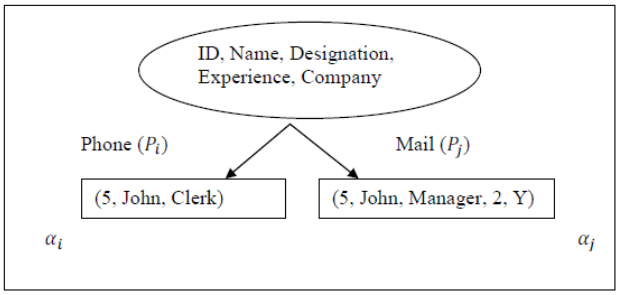

Figure 2 Topic-based Pattern Tree representation

From the above figure, the Topic-based Pattern Tree representation includes a root node 'r' with a pattern 'P'. The results of the two patterns 'P_i' and 'P_j' are evaluated and stored as resultant 'α_i' and 'α_j'. Let us consider two patterns 'P_i' and 'P_j' that retrieves information about the employee details of a company be mathematically formulated as given below

$$\alpha_i \rightarrow P_i = (ID, Name, Designation) \quad (2)$$

$$\alpha_j \rightarrow P_j = (ID, Name, Experience, Company) \quad (3)$$

From (2) and (3), two patterns 'P_i' and 'P_j' are used to retrieve the information based on relevance factor, aiming to reduce the query retrieval time. The relevance factor applied for measuring the query retrieval time is given as below

$$RF = \sum_{i,j=1}^{n} Rel\ Max\ (P_i, P_j) \quad (4)$$

The relevance factor 'RF' measures the relevance of the patterns 'P_i' and 'P_j' with respect to the user query '$Query_i$'. The function '' measures the maximum relative response between the patterns and stores the resultant value as the relevance factor, therefore ensuring the query retrieval time.

Design of Probability-based Latent Dirichlet Allocation

The second step involved in the design of PT-LDA framework is the design of Probability-based Latent Dirichlet Allocation.

The objective behind the design of Probability-based Latent Dirichlet Allocation is to improve the query retrieval ratio through mapping. By efficiently mapping in the patterns using the relevance factor PT-LDA framework, the query retrieval ratio is improved in a significant manner.

The Probability-based Latent Dirichlet Allocation in PT-LDA framework performs an efficient mapping between 'α_i *and* α_j' if the following condition is satisfied

$$Prob\ ((\alpha_i) \cap\ Prob\ (\alpha_i)) = Res\ (\alpha_i\ \cap\ \alpha_j) \quad (5)$$

From (5), several patterns with each patterns corresponding to a user query is characterized by its own distribution over the topics 'T_i' in the document 'Doc'. This in turn reduces query retrieval time and therefore query retrieval ratio. Figure 3 given below shows the Pattern Tree-based LDA algorithm

Input: document 'Doc', topics '$T_i = T_1, T_2, \ldots, T_n$', words '$w_i = w_1, w_2, \ldots, w_n$', query '$Query_i = Query_1, Query_2, \ldots, Query_n$', patterns '$P_i$' and '$P_j$'
Output: Efficient query retrieval
Step 1: Begin Step 2: For each document 'Doc' and query '$Query_i$' Step 3: Repeat Step 4: Derive the resultant patterns using (2) and (3) Step 5: Measure relevance factor using (4) Step 5: Perform mapping using (5) Step 6: Until (all patterns are used for query retrieval) Step 7: End for Step 8: End

Figure 3 Pattern Tree-based LDA algorithm

The figure given above shows the Pattern Tree-based LDA algorithm which includes two steps. For each document, the first step derives the patterns for each query given by the user. The second step performs mapping through probability measure using LDA model based on the pattern tree root node. This helps in retrieval of the results in a significant manner reducing the query retrieval time and retrieval ratio.

Experimental settings

The Pattern Tree-based Latent Dirichlet Allocation (PT-LDA) framework uses JAVA platform to perform the experimental work. The PL-LDA framework performed in JAVA platform uses the Reuters-21578 Text Categorization Collection Data Set from UCI repository to identify the query retrieval rate. Reuters-21578 has a collection of documents which were collected and indexed based on categories.

Each REUTERS tag includes five attributes, such as TOPICS, LEWISSPLIT, CGISPLIT, OLDID, and NEWID. Using these attributes the document topic name and association between the documents are identified in an efficient manner. To measure the efficacy of the PT-LDA framework, the work is compared against the existing Rule based method using Semantic Relation using Automatic Enrichment (SR-AE) [1] and Automatic Discovery of Personal Name (ADPN) [2]. Experiment is conducted on the factors such as query retrieval time, query retrieval ratio and accuracy of query being retrieved.

The query retrieval time is the time taken to retrieve the result with respect to the number of queries issued by the user. The query retrieval time is measured in terms of milliseconds and is mathematically evaluated as given below.

$$QRT = n * Time\ (retrieve\ result) \quad (6)$$

From (6), the query retrieval time 'QRT' is measured using the queries 'n'. Lower the query retrieval time, more efficient the method is said to be. The query retrieval ratio is the ratio of successful resultant retrieval of queries to the total number of queries. The query retrieval ratio is formulated as given below.

$$QRR = \left(\frac{Succesful\ retrieval}{n}\right) * 100 \quad (7)$$

From (7), the query retrieval ratio 'QRR' is measured in terms of percentage (%). Higher the query retrieval ratio more efficient the method is said to be. Processing time refers to the time taken to perform efficient mapping between 'α_i *and* α_j' with respect to 'n' number of queries and is formulated as given below.

$$PT = n * Time\ (Res\ (\alpha_i\ \cap\ \alpha_j)) \quad (8)$$

From (8), the processing time 'PT' is measured in an extensive manner for any number of queries. Lower the processing time, more efficient the method is said to be.

Discussion

In order to measure the efficiency of Pattern Tree-based Latent Dirichlet Allocation (PT-LDA) framework, the performance was accessed using five attributes. The five attributes used to measure the efficacy of PT-LDA framework are TOPICS, LEWISSPLIT, CGISPLIT, OLDID, and NEWID obtained from Reuters-21578 Text Categorization Collection Data Set from UCI repository. They are compared and analyzed with the Semantic Relation using Automatic Enrichment (SR-AE) [1] and Automatic Discovery of Personal Name (ADPN) [2]. The experimental results using JAVA are compared and analyzed with the help of the table values and graphical representation is shown as given below.

A. Scenario 1: Query Retrieval Time

To deliver with a detailed performance, in Table 1 we apply Pattern Tree-based LDA algorithm to obtain the query retrieval time and comparison is made with two other existing techniques, SR-AE and ADPN respectively. Query retrieval time in PT-LDA framework refers to the rate at which the queries are retrieved with respect to the number of queries. Lower query retrieval time results in the improvement of the framework.

Figure 4 show that the proposed PT-LDA framework provides lower query retrieval time when compared to SR-AE [1] and ADPN [2] with respect to the number of queries provided as input. This is because of the application of Topic-based Pattern Tree using which the queries are retrieved in an efficient manner which vehemently results in the improvement of query retrieval time 14.78% compared to SR-AE. In addition to that with the use of Topic-based pattern

tree representation relevance factor are applied to measure the query retrieval effectiveness and therefore an improvement in query retrieval time is observed using PT-L by 17.65% compared to ADPN respectively.

Table 1 Tabulation for query retrieval time

No. of queries (n)	Query retrieval time (ms)		
	PT-LDA	SR-AE	ADPN
10	3.85	4.88	5.08
20	5.28	6.31	6.51
30	4.85	5.88	6.08
40	9.29	10.32	10.52
50	12.32	13.35	13.55
60	10.24	11.27	11.47
70	15.86	16.89	17.09

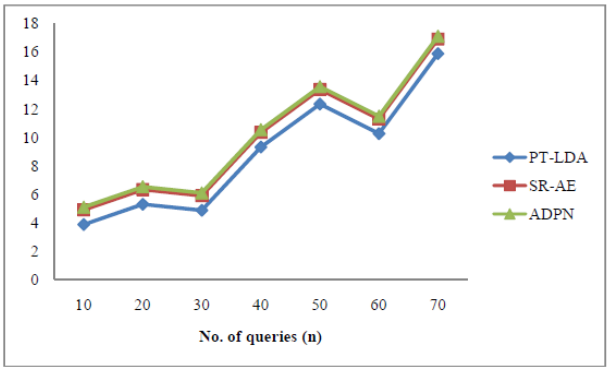

Figure 4 Impact of query retrieval time with respect to queries

B. Scenario 2: Query Retrieval Ratio

The comparison of query retrieval ratio is provided in table 2 with respect to different queries in the range of 10 – 70. With increase in the number of queries being issued by the user at varied time period, the query retrieval ratio is also increased though reaches to saturation when 30 queries were issued and a drift change occurred when the queries issued was observed to be 70. This is because the Reuters-21578 Text Categorization Collection Data Set used contains the text retrieved at different time which is not same throughput the day. As a result, the variations are also noted.

Table 2 Tabulation for query retrieval ratio

No. of queries (n)	Query retrieval ratio (%)		
	PT-LDA	SR-AE	ADPN
10	58.35	50.30	44.24
20	65.21	57.16	51.10
30	71.35	63.30	57.24
40	64.29	56.24	50.18
50	73.44	65.39	59.33
60	75.82	67.75	61.69
70	80.27	72.22	66.16

In figure 5, the query retrieval ratio is depicted with respect to several observations of queries range 10 – 70 that are considered for the experimental purposes in JAVA. From the figure, we can observe that the value of query retrieval ratio achieved using the proposed PT-LDA framework is higher when compared to two other existing works Semantic Relation using Automatic Enrichment (SR-AE) [1] and Automatic Discovery of Personal Name (ADPN) [2]. Moreover, it is also observed that by changing the number of observations being made, the query retrieval ratio is increased but comparatively improvement is observed using the proposed PT-LDA. This is because with the application of Probability-based Latent Dirichlet Allocation, probability measure is used to obtain the query retrieval rate. Moreover with the efficient mapping function applied in the PT-LDA framework, there results in the improvement of query retrieval ratio by 11.65% compared to SR-AE and 20.42% compared to ADPN respectively.

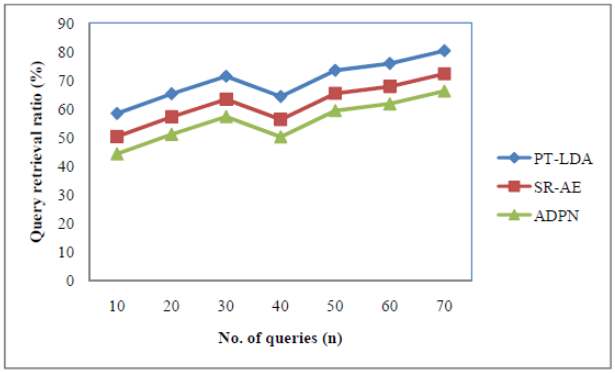

Figure 5 Impact of query retrieval ratio

C. Scenario 3: Processing time

The effectiveness on processing time measures the rate at which the analysis between the patterns and their processing time is made using Pattern Tree-based LDA algorithm. The Pattern Tree-based LDA algorithm using PT-LDA framework is provided in an elaborate manner in table 3. We consider the approach with different number of queries acquired from the UCI repository using four attributes for experimental purpose using JAVA.

The targeting results of the query processing time using PT-LDA framework with two state-of-the-art methods [1], [2] in figure 6 is presented for visual comparison based on the varied queries. Our framework differs from the SR-AE [1] and ADPN [2] in that we have incorporated the Pattern Tree-based LDA algorithm for handling different queries arrived at different time period reduces the processing time by 29.36% and 48.97% compared to ADPN respectively. By applying two different elements, called probability measure and efficient mapping between the patterns for each queries, efficient analysis of relativity is performed that results in the improvement of effectiveness on query retrieval ratio and therefore reduces the processing time.

Table 3 Tabulation for processing time

No. of queries (n)	Processing time (ms)		
	PT-LDA	SR-AE	ADPN
10	12.35	18.40	22.48
20	15.89	21.94	25.99
30	20.31	26.36	30.38
40	24.87	30.92	34.97
50	22.19	28.24	32.23
60	30.32	36.37	40.39
70	35.19	41.24	45.26

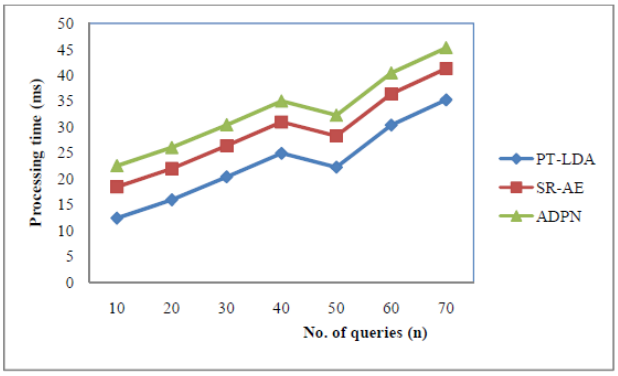

Figure 6 Impact of processing time

Conclusion

In this work, a Pattern Tree-based Latent Dirichlet Allocation (PT-LDA) framework is designed to ensure query retrieval ratio consistency on Reuters-21578 Text Categorization Collection Data Set. The PT-LDA framework utilizes Topic-based Pattern Tree model and Probability-based Latent Dirichlet Allocation model to retrieve the user query and achieve optimization in mining. With the Topic-based Pattern Tree model simplify query retrieval where the results of complex queries are addressed in an efficient manner with the aid of pattern tree model. As a result, the pattern tree structure helps in minimizing the query retrieval time. Next, with the introduction of Probability-based Latent Dirichlet Allocation model, efficient query retrieval is performed using probability measure. Finally, with the application of Pattern Tree-based LDA algorithm, consistent and flexible queries are retrieved resulting in the efficient processing of queries and therefore reducing the processing time in a significant manner. Experimental results demonstrate that the proposed PT-LDA framework not only leads to noticeable improvement over the parameters query retrieval time and query retrieval ratio, but also but also outperforms query processing time compared over other state-of-the-art works.

References

[1] Myunggwon Hwang, Chang Choi and Pankoo Kim,“ Automatic Enrichment of Semantic Relation Network and Its Application to Word Sense Disambiguation,” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 6, JUNE 2011

[2] Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka,“Automatic Discovery of Personal Name Aliases from the Web,” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 6, JUNE 2011

[3] Jieru Cheng, Wenjun Wang, Rui Gao,” Massive RDF Data Complicated Query Optimization Based on MapReduce”, Elsevier, 2012 International Conference on Solid State Devices and Materials Science

[4] Jihyun Lee, Jun-Ki Min, Alice Oh, Chin-Wan Chung,” Effective ranking and search techniques for Web resources considering semantic relationships”, Information Processing and Management, Elsevier, Sep 2013

[5] Gideon Zenz a, Xuan Zhou b, Enrico Minack a, Wolf Siberski a, Wolfgang Nejdl,” From Keywords to Semantic Queries | Incremental Query Construction on the Semantic Web”, Elsevier, July 2009

[6] Zhifeng Bao, Jiaheng Lu, Tok Wang Ling, and Bo Chen,”Towards an Effective XML Keyword Search”,IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 8, AUGUST 2010

[7] Mihaela A. Bornea, Vasilis Vassalos, Yannis Kotidis, and Antonios Deligiannakis,” Adaptive Join Operators for Result Rate Optimization on Streaming Inputs”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 8, AUGUST 2010

[8] Weifeng Su, Jiying Wang, and Frederick H. Lochovsky,” Record Matching over Query Results from Multiple Web Databases”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 4, APRIL 2010

[9] Boris Motik , Ian Horrocksa, Ulrike Sattler,” Bridging the gap between OWL and relational databases”, Web Semantics: Science, Services and Agents on theWorldWideWeb, Elsevier, Mar 2009

[10] Tak-Lam Wong and Wai Lam,” Learning to Adapt Web Information Extraction Knowledge and Discovering New Attributes via a Bayesian Approach”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 4, APRIL 2010

[11] Xiaomin Ning, Hai Jin, Weijia Jia, Pingpeng Yuan,” Practical and effective IR-style keyword search over semantic web”, Information Processing and Management, Elseiver, Oct 2009

[12] Bettina Fazzinga, Giorgio Gianforme, Georg Gottlob, and Thomas Lukasiewicz,” Semantic Web Search Based on Ontological Conjunctive Queries”, Elsevier, May 2011