Semantic Information Retrieval based on Topic Modeling and Community Interests Mining

Minuri Rajapaksha
Department of Computational Mathematics
University of Moratuwa
Sri Lanka
rpmchathu@gmail.com

Thushari Silva
Department of Computational Mathematics
University of Moratuwa
Sri Lanka
thusharip@uom.lk

***Abstract*— Search engines or localized software systems developed for information searching, play an important role in knowledge discovery. Proliferation of data in the web and social media has posed significant challenges in finding relevant information efficiently even using those search engines or other software systems. Moreover, those systems or engines tend to collect large number of data, which could be useful for end users in various ways but have overlooked the meaning of the search phrases, hence generate irrelevant search results. A unit level searching i.e. searching information within a website or page is also not effective as they follow exact keyword matching techniques and ignore the semantic level matching of search phrases. In order to address those deficiencies, this research proposes a hybrid approach which use the semantics of data, community preferences as well as collaborative filtering techniques for semantic information retrieval. More specifically, Topic modeling based on Latent Dirichlet Allocation together with topic-driven based community detection methods are applied for identifying personalized search results generation and hence improve the relatedness of the research results. Based on the proposed hybrid approach a framework for semantic search that can easily be integrated to a software application has been implemented. The evaluation results confirm the effectiveness of search results which outperform benchmark approaches that follow traditional keyword search algorithms.**

Keywords*— *Semantic data mining, Latent Dirichlet Allocation, collaborative filtering, community detection, semantic information

I. INTRODUCTION

There is a great growing demand for sophisticated information retrieval using search engines that accommodate various needs of different users [1]. In order to meet this demand current systems heavily used several data representation techniques in computer systems including database models, especially relational databases which enables efficient information storing and querying [2]. Since most of the current web applications are supporting relational schema-based information retrieval, they could not provide any semantic sense of feedback to users. Ontologies emerged as an alternative to databases in such applications that require a more 'enriched' meaning [3].

Todays web applications need to utilize huge amount of data and users of those applications are looking for efficient accesses to the data that they want. Moreover, those applications contain large number of documents and handling such documents and retrieval of information should be very efficient for obtaining a real business value. Almost all search engines use text-based searching techniques in which the query string is matched with the text in files or a database. Then the search results are generated just based on the number of occurrences of the string and this does not take real meaning i.e. semantics of the query string. The same applies to an application where a user tries to find help details inside an application. Hence searching based on content only has become a challenge to most applications [4].

One of the primary applications of knowledge extraction is automatic extraction of topics discussed by people from large volumes of texts. Some examples of large text are feeds in social media, reviews, news stories, e-mails of customer complaints etc... Knowing what people are talking about and understanding their opinions is highly valuable to businesses, administrators, political campaigns. It is really hard to manually go through such large volumes and compile the topics. Thus, it is required an automated algorithm that can read through the text documents and automatically generate the topics discussed by considering the semantics of the topics. In such a case, semantic search strategies could enhance semantic search results.

Search engines including Google, yahoo Bing, and DuckDuckGo provide search results with personalization up to a certain extent, but in general most of individual sites lack of semantic search mechanism even within its site. Intelligent Semantic Frameworks for individual sites [5] has focused on Latent Semantic Indexing and personalization based on users' history. Current search engines lack a model that focus on semantics of data. Moreover, these search techniques have overlooked preference of the searching community that could be learned through their searching patterns. Thus, most of current approaches have largely ignored the dynamics of searching phrases of individuals and the semantics of data.

In order to meet these deficiencies, this paper presents a model focused more on semantics of data extraction based on LDA (Latent Dirichlet Allocation) model and community detection method. Further, the model could optimize results in a way that they meet all user needs. Thus, the proposed solution could also serve as an internal semantic search engine that can be used within its site.

The remaining part of the paper is organized as follows. Section II presents the previous work on semantic meta data extraction, community detection and collaborative filtering. Section III introduces the LDA model and presents the proposed approach. Section IV presents the implementation details of the system. Section V demonstrates the experimental results. Section VI concludes the paper and outlines the future work.

978-1-7281-3632-5/19/$31.00 ©2019 IEEE

II. Related Work

Present web applications related to information retrieval, which utilize search engines, is lack of pluggable search engines [27]. Some present semantic search engines are depending on other search engines and they are not supporting all browsers as presented in Table I. Thus, they do not provide semantically rich and personalized search results for dynamic queries. As an alternative, some applications which are based on search engines use user history and already established communities such as follower network topology in Twitter [6]. Detecting communities dynamically from the semantic content itself is hardly found in literature. Semantic search engines lack of dynamic community detection mechanism which uses the current semantics of data, hence it is hard to find personalized interest of the community. Following subsections present details about topic modeling, community detection methods and collaborative filtering.

TABLE I. Semantic Search Engines Current Practices and Issues

Semantic web engine	Approach	Technique	Features	Limitations
Hakia [31]	Related searches, NLP	OntoSem, QDEX	Excellent resumes, Easily Identifies Information from credible sites, Saves time	Does not index everything. It needs other search engines.
Swoogle	Content based	A crawler-based indexing semantic search engine that searches ontologies and instance data	Finds appropriate ontologies, appropriate instance data structures of semantic web	Extending Swoogle to index and effectively query large amounts of instance data is still a challenge
Lexxe [32]	NLP	It uses semantic key technology which enable users to query with a conceptual keyword	Part-of-speech tagging, Parsing, Word sense disambiguation	It does not work well with long queries
Factbites [33]	Contextual search	AI and computational linguistics	It has the ability to filter out spam websites in the search results. It searches based on topic rather than keywords.	It works better with general questions rather than specified topics.
Duckduckgo	Clustered search, NLP	Instant answers are collected from either 3rd party APIs or static data sources like text files.	Zero click information, emphasizes privacy and does not record user information, produces result based on many sources and its own web crawler	It lacks feature for image and video searching
SenseBot [34]	ConceptSearch	Identifies key semantic concept from user's query by using text mining algorithm that parse the web pages which are then used to perform coherent summary	Multi-document summarization	It only works with Firefox as a browser extension and Google search engine to display results
Kngine	Knowledge-based approach and statistical tool	Knowledge base determines synonyms, relations between concepts, meaning document analysis and context based fuzzy search	It displays search results in the form of images, it is multi-lingual, has the ability to allow the users to search in parallel manner	It does not provide silent mode option

A. Topic Modeling

The predecessor of topic modeling can be traced back to LSA (Latent Semantic Analysis) [7]. LSA is based on spatial dictionaries. Implicit semantic documents in LSA are implemented in low-dimensional representation of space, but this representation does not support for the problem of the coexistence of many possible meanings for a word or phrase. Hofmann proposed Probabilistic Latent Semantic Analysis (PLSA) [8,9] for the defect of LSA, mainly using the probability distribution corresponding to one dictionary in each dimension. However, PLSA does not provide a probabilistic model at the document level. This leads to overfitting problems due to linear increment in the number of parameters to be estimated in the model, with the size of the corpus. LDA (Latent Dirichlet Allocation) (Fig. 1) [10] is a generic model that uses the Dirichlet priori distribution of topics to overcome the shortcomings of PLSA. This model can find the semantic structure of the text set. Current web applications based on semantic search engines hardly use latest semantics of data extraction technologies. Most of search engines are implemented based on LSA [5].

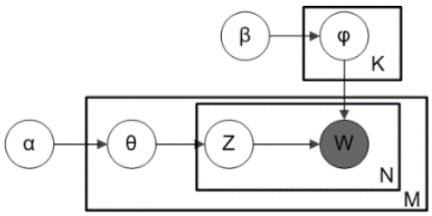

Fig. 1. Latent Dirichlet Allocation model

B. Community Detection

Community detection algorithms have been widely used for detecting hidden communities. They can be generally divided into two classes: topic-based approaches and structure-based approaches. The structure-based approach [11,12] does not have a clear perspective on how to make sense of identified communities. Only a few research efforts fall into the topic-based approach, which groups individuals sharing common topic interests into a community.

Reference [18] proposed a generative model to discover communities based on topics, social graph topology, and nature of user interactions. Reference [19] proposed a Bayesian generative model for community extraction which considered both the network topology and user topics to generate communities. Structure-based approaches for detecting communities mine shared common interests on Twitter based on their relationship hierarchies starting from celebrities which represent an interest category [20]. A LDA-based model has been used to detect user topics based on their posted tweets [21].In this approach a weighted topic graph also called semantic graph is constructed. The weight of an edge is corresponding to the topic similarity of two users who are nodes connected by the edge. Once the topics graph is created a community detection algorithm is applied to find out the community in the topic graph. Reference [22] proposed an approach based on grouping users who share the same interests mined through their textual posts. This method follows Principal Component Analysis to find the principle components, called interest centers and uses K-mean clustering algorithm to cluster the users based on their distance to principle components. Current semantic search engines lack a dynamic approach which uses the user uploaded files for detecting communities.

C. Collaborative Filtering

Personalization is essential for personalized search results suggestion. Providing search results based on user preferences as well as their profiles is one way of doing personalization and it is tightly coupled with the relevant search functionality. Collaborative filtering approach which is widely used in recommendation systems, is used for deriving profiles based on historical events [13]. Current semantic search engines' personalization is based on both user history and predefined

communities, but they have overlooked dynamic detection of communities. Item based collaborative filtering [14] has overcome scalability problems and literature emphasizes is better than K-nearest algorithm.

III. SEMANTIC SEARCHING FRAMEWORK

This section presents the architecture of the Semantic Searching framework.

A. Architecture of Semantic Searching Framework

Fig. 2 illustrates the top-level architecture of the system. The system contains five modules called UI module, Discover topics Module, Community extraction module, Item based collaborative filtering Module and Search optimization Module. The system proposes a hybrid approach which uses semantics of data, community preferences as well as collaborative filtering techniques for semantic information retrieval. Moreover, the proposed system uses existing LDA topic model, Jesen-Shannon Divergence algorithm and item based collaborative filtering algorithm. The novelty of this research lies on the proposed hybrid approach which integrates, topic modeling, dynamic community analysis and interest mining through collaborative filtering approach for semantic information retrieval.

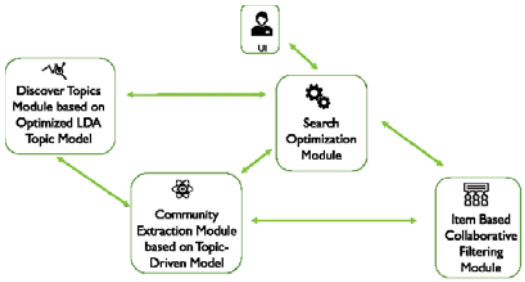

Fig. 2. The proposed approach for Semantic Search

1) Discovery of Topics Based on Optimized LDA Topic Model

The module uses the Latent Dirichlet Allocation (LDA)for discovering hidden topics from the articles. . Fig. 3 demonstrates the inputs and outputs of LDA model.

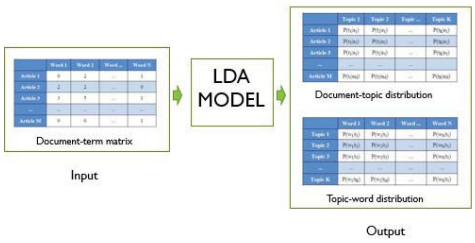

Fig. 3. Latent Dirichlet Allocation Model – Input and Output Behaviour

For optimizing LDA model, we computed Model Perplexity as well as Coherence Score. The Coherence Score provides a convenient measure to judge how good a given LDA topic model is. Perplexity lower is the better. coherence higher is the better. We built many LDA models with different values of number of topics (k) and picked the one that gives the highest coherence value.

2) Community Extraction Based on Topic-Driven Model

By using LDA topic-model which return user-topic distributions and Jensen-Shannon Divergence algorithm [23], the topic distances between two users have been calculated. Then using the calculated distances, community graph has been constructed. In order to construct user communities an approach of Girvan-Newman which is based on divisive classification is followed. The above steps have been explained below.

a) Calculate Distance Between Users

The distance between user i and user j is computed by using the Jensen-Shannon Divergence between the topic distributions on user i and j [23] and it can be computed as follows (1).

$$dist_T(i,j) = \sqrt{2 * D_{JS}(i,j)} \tag{1}$$

, where $D_{JS}(i,j)$: the Jensen-Shannon Divergence between the two topic distributions DTi and DTj. It is defined as (2).

$$D_{JS}(i,j) = \frac{1}{2}(D_{KL}(DT_i||M) + D_{KL}(DT_j||M)) \tag{2}$$

, where M: the average of the two probability distributions. DKL: the Kullback-Leibler Divergence which defines the divergence from distribution Q to distribution P is computed as follows.

$$D_{KL}(P||Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} \tag{3}$$

As the first measure (domains), we calculate the distance between users as the Jensen-Shannon Divergence between domains distributions over users as in formula 1 and 2. The second measure (topics-domains) combines the two previous measures (Topics, Domains). This new measure allows to decrease the distance between users who do not show any similarity on the topics but also in the same domains. The distance between users in this measure is computed as follows.

$$dist_{TD}(i,j) = \sqrt{2 * D_{JS}(i,j)} \tag{4}$$

b) Construction of the Semantic Graph

In the semantic graph nodes and edges represent users and topic distance between users respectively. In the topic graph, we create an edge from the user *i* to the user *j*, if the user *j* is closest to the user *i* for a topic *k*, the weight of this link is calculated as the distance between them for the selected topic *k*. Moreover, if there is another edge from the user *i* to the user *j* for another topic, it is enough to choose the edge with minimal distance between these two users *i* and *j*.

c) Construction of User Communities

Girvan-Newman approach which is based on divisive classification is used for the construction of user communities. Girvan-Newman Algorithm is as follows.

1. Initially, assume that all nodes are in one community

2. Calculate betweenness scores for all edges

3. Find an edge with highest betweenness score and remove it from network

4. Recalculate betweenness for all remaining edges

5. Remove edges until we get all communities

Anthropologist Dunbar [24] suggests that the size of communities with strong ties in both traditional social networks and Internet-based social networks should be limited to 150 (called Dunbar's number) because of human's cognitive constraints and time constraints. Large communities of size over 150 people contain weak connections among their members and are therefore not stable. Therefore the community size was limited to 150 users.

3) Item Based Collaborative Filtering Module

Item based collaborative filtering considers about the user search history in a particular community (based on communities identified by above community detection module) . Aaccording to the frequency of any article viewed or downloaded, an article is given a preference value. These inferred preference values are stored with the search results and they are used to derive a final composite score, on which ultimate search results are based on.

Pair wise similarities of the columns of the rating matrix is computed and top 20 to 50 most similar items for a given article in the item-similarity matrix is found. In order to generate predictions, we computed a weighted sum over all articles similar to the unknown/given article that have been rated by the current user (5). The output of this module is a list of recommended articles for particular user.

$$p_{ui} = \frac{\sum_{j \in S(i,u)} s_{ij} r_{uj}}{\sum_{j \in S(i,u)} |s_{ij}|} \quad (5)$$

4) Search Optimization Module

This module is responsible for integrating above modules to search engine, build indexing, rank files, execute the query and find best search results.

Main output called document-topic distribution and topic-word distribution in Discovery of topics module is used to calculate the distance between users in Community extraction module. Calculated user topic distributions and the distance between users are used to construct the graph. The graph information is stored in a CSV file Above generated CSV file in Community extraction module is used to construct user communities. Community extraction module returned output of List of communities with belonging users (CSV file) and the users history is passed as an input to the item based collaborative filtering module. Item based collaborative filtering module will return ranked articles considering community interests and user history. Finally search optimization module consider the ranked files returned from collaborative filtering module for optimizing ranking and Lucene-LDA model for indexing and returning search results.

Genetic algorithm (GA) is used to optimize the search results. Genetic algorithm is a search heuristic that mimics the process of natural evolution and is thus routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as mutation, selection, and crossover. [28, 29, 30]. There are 2 populations of chromosomes to be evolved.

- each chromosome is composed of a pre-defined number of words randomly defined randomly chosen from the keywords library of each folder.
- contains the same number of genes of the 1st population and same number of individuals of the previous operation. Each gene of the 2nd population may assume 1 of the 3 Boolean random values: AND, OR, NOT.

In Genetic algorithm a particular chromosome may be ranked against all the other chromosomes. Optimal chromosomes are allowed to breed and mix their datasets producing a new generation that will be even better.

After determining the fitness of all the individuals, a binary tournament is performed to select those individuals that will compose the next generation. The best individual of the population is maintained and is not subjected to crossover.

The crossover operator implemented here was the single-point crossover. The only constraint being that the same word cannot appear twice in the same chromosome in which case, the crossover operator is not applied and the parent chromosomes remain unchanged. Hence applying genetic algorithm searching process has been optimized.

IV. IMPLEMENTATION

The overall software has been developed as an ASP .Net web application. However different modules inside the system have been implemented with different technologies such as Python programming language using large number of libraries like Gensim, NLTK, MALLET etc, ANACONDA platform for easy source code maintenance and Lucene has used as Search Engine platform for indexing and integrating other modules.

Lucene is used to internalize the topics and topic memberships while building the index and executing the queries [25]. List of terms in the corpus (i.e. term list), matrix that specifies the membership of each word in each topic. (topic-word distribution), matrix that lists the original file names that LDA was executed on and matrix that specifies the topic membership of each file in each topic (document-topic distribution) are the input files to system.

In this research we used Payloads to cleverly encode the topics in each document at index time. When user has entered the query, system determines which topics are in the query. Then create a Payload query based on these topics. Lucene will then find all documents that contain these topics. We ignore the actual relevancy returned by Lucene, and instead use the contents of the Payload to compute the relevancy ourselves, and re-rank the results.

V. EXPERIMENTAL RESULTS

This section presents the observations on the discovered topic accuracy and topic similarities of users in dynamically defined communities. There were 20 cooperative participants who were serving as web application users. The eexperiment was done using personal computers, in which python libraries are installed.

A. Training Data

The corpus used to train an LDA model, is a collection of articles in English Wikipedia which was downloaded from [26] in September 2018. The collection consists of over 5 million articles. Articles are preprocessed with the removal of

unnecessary words including stop words, URLs, articles, file attachments, XML labels, special characters, digits, spaces, new lines, punctuations etc... It Uses lemmatization for further filtering of necessary data. Lemmatization is converting a word to its root word. For example: the lemma of the word 'machines' is 'machine'. Likewise, 'walking' –> 'walk', 'mice' –> 'mouse' and so on. We kept only articles with more than 150 English characters. This gives us a corpus with a list of document ids, word frequency-dictionary and a list of words with their ids. All the tokens(i.e.words) in the dictionary which either have occurred in less than 4% articles or have occurred in more than 40% of the articles are removed from the dictionary.

B. LDA Model Performance

LDA model is trained in three iterations by setting the number of topics to 10, 20, and 40 respectively. Fig. 4 illustrates computed coherence score when number of topics(K) equals to 10, 20 and 40. In LDA model higher the coherence score means the trained topic model is more accurate. It can be justified when compare with Table II. Table II interprets the actual topics defined for given articles for different K values along with human prediction. When comparing those two results we can interpret that the best LDA model creates when K=20 since it is closer to topics defined by human.

TABLE II. EVALUATE THE MODEL BASED ON DIFFERENT TOPIC NUMBERS

A	K = 10	K = 20	K = 40	Human
1	Government	War	Game	War
2	People	Education_Research	Literature_or_Study	Education
3	People	Movie_TV_Show	Music	Music
4	Award_Ceremony	Games	People	Sports
5	Space_Astronomes	Weather_or_Sport	Natural Disaster	Weather
6	unknown	Engineering	Engineering	Computer
7	LandScapes	Nature	Natural Species	Biology
8	Governing	Finance	unknown	Business
9	Country	Computer	Software_Computer	Engineering
10	Relationships	Food_Meals	unknown	Food

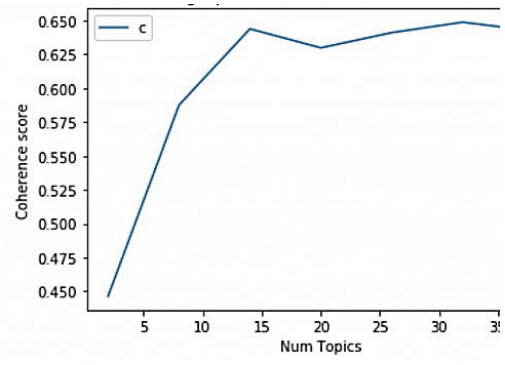

Fig. 4. Choosing optimal model for Latent Dirichlet Allocation model using coherence scores

C. Community Detection

Calculate distance between users using Jensen Shannon Divergence (Table III).

TABLE III. DISTANCE BETWEEN USERS

	User1	User2	User3	User4	User5
User1	0.00000	1.260869	1.163385	10226157	1.243853
User2	1.260869	0.00000	1.231731	1.264484	1.247072
User3	1.163385	1.231731	0.00000	1.263051	1.142907
User4	1.226157	1.264484	1.263051	0.00000	1.230949
User5	1.243853	1.247072	1.142907	1.230949	0.00000

Construct graph (Fig. 5) based on closeness between users according to selected topic (Table IV).

TABLE IV. CLOSENESS BETWEEN USERS FOR SELECTED TOPIC

	Topic1	Topic2	Topic3	Topic4	Topic5
User1	0.016541	0.001504	0.001504	0.978947	0.001504
User2	0.001835	0.001835	0.992661	0.001835	0.001835
User3	0.952838	0.000873	0.000873	0.044541	0.000873
User4	0.000608	0.006687	0.000608	0.012766	0.979331
User5	0.079154	0.906949	0.006647	0.000604	0.006647

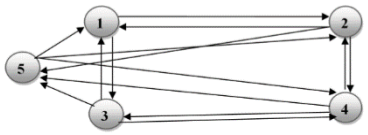

Fig. 5. Constructed graph based on closeness between users

Fig. 6 illustrates first five most related topics in one community. Here the majority of users in this community treat topic "Music". When compare with actual group of users in this detected community, interest of actual community also closer to "Music".

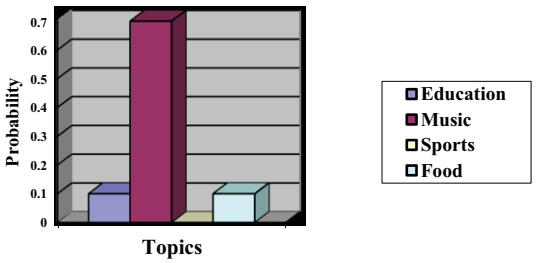

Fig. 6. Topics treated in community1

D. Search Optimization

TABLE V. FILTER ARTICLES BASED ON ITEM BASED COLLABORATIVE FILTERING

Article	Probability	Tag
Article_19	f(Article_19)/n(articles) = 0.01	Collaborative
Article_29	f(Article_29)/n(articles) = 0.06	Collaborative
Article_1	f(Article_1)/n(articles) = 0.04	Collaborative

1. Filter using Item-based collaborative filtering (Table V)

Take top 3 user belonging clusters and consider users of those clusters. Consider above users' search history and filter

related articles from user uploaded documents with probabilities for given search query.

VI. Conclusion and Future Work

It is evident that the search engine framework presented in this research work can perform better in searching than a conventional search engine. It enables users to identify the community preferences even when there is no pre-established community topologies. Therefore, the users of internal web applications who do not have pre-defined communities, could access the relevant information which is hidden in large volumes of data without any usual hassle of searching.

This framework can be integrated to individual web sites and would remarkably improve relevant search results. Furthermore, integration of personalization based on dynamically defined communities will be useful for users.

The system is performing well for defining topics for given articles and cluster user communities. The system only supports for PDF, Word, XML documents and that is one of the limitations of the system. Hence topic modeling and topic-driven approach to detect dynamic communities can provide semantically rich search engine framework to provide best search results with personalization for a given query.

ACKNOWLEDGMENT

This research work is funded by SRC grant: SRC/LT/2017/15 granted by University of Moratuwa.

References

[1] R. Tehseen, "Semantic Information Retrieval: A Survey," Journal of Information Technology and Software Engineering, vol. 8, p. 241, 2018

[2] M. Yadav, "A Review Paper on Information Retrieval Techniques for Point and Range Query in Database System," International Journal of Advanced Research in Computer Science, p. 5, 2017.

[3] M. Bansal and J. Arora, "A Review on Ontology Based Information Retrieval System," International Journal of Engineering Development and Research, vol. 4(2), pp. 263-265, 2016.

[4] E. A. Calvillo, A. Padilla, J. Munoz, J. Ponce and J. T. Fernandez, "Searching research papers using clustering and text mining," in 23rd International Conference on Electronics, Communications and Computing, 2013.

[5] M. Jayaratne, I. Haththotuwa, C. Dandeniya Arachchi, S. Perera, D. Fernando and S. Weerakoon, "iSeS: Intelligent Semantic Search Framework," in Proceedings of the 6th Euro American Conference on Telematics and Information Systems EATIS, 2012

[6] B. Dib, F. Kalloubi, E. Nfaoui and A. Boulaalam, "Semantic-based Followee Recommendations on Twitter Network," in Procedia Computer Science, 2018.

[7] S. Deerwester, S. T. Dumais, G. Furnas, T. K. Landauer and R. Harshman, "Indexing by latent semantic analysis," Journal of the Association for Information Science and Technology, vol. 41, no. 6, pp. 391-407, 1990.

[8] S. Deerwester, "Indexing by latent semantic analysis," Journal of the Association for Information Science and Technology, vol. 41, no. 6, pp. 391-407, 2010.

[9] T. Hofmann, "Unsupervised learning by probabilistic Latent Semantic Analysis," Machine Learning, vol. 42, no. 1-2, pp. 177-196, 2001.

[10] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation," Journal of Machine Learning Research, vol. 3, pp. 993-1022, 2003.

[11] W. Ding and Zhaoyun, "Mining user interest in microblogs with a user-topic model," China Communications, vol. 11, no. 8, pp. 131-144, 2014.

[12] T. Hofmann, "Probabilistic latent semantic indexing," in Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99), Berkeley, Calif, USA, 1999.

[13] R. Chen, Q. Hua, Y. Chang, B. Wei, L. Zhang and X. Kong, "A survey of collaborative filtering-based recommender systems," From traditional methods to hybrid methods based on social networks, vol. 6, pp. 64301-64320, 2018.

[14] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the tenth international conference on World Wide Web - WWW, Hong Kong, 2001.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan, ''Latent Dirichlet allocation,'' J. Mach. Learn. Res., vol. 3, pp. 993–1022, Mar. 2003.

[16] D. M. Blei, "Probabilistic topic models," Communications of the ACM, vol. 55, pp. 77-84, 2012.

[17] D. Lemire and A. Maclachlan, "Slope One Predictors for Online Rating-Based Collaborative Filtering," in Proceedings of the 2005 SIAM International Conference on Data Mining, 2005.

[18] M. Sachan, D. Contractor, T. A. Faruquie and L.V.Subramaniam, "Using content and interactions for discovering communities in social networks," in Proceedings of 21st International Conference World Wide Web, 2012.

[19] N. Pathak, C. DeLong, A. Banerjee and K. Erickson, "Social topic models for community extraction," in Proceedings of The 2nd SNA-KDD workshop, 2008.

[20] K. Lim and A. Datta, "A Topological Approach for Detecting Twitter Communities with Common Interests," in Proceedings of Ubiquitous Social Media Analysis, Berlin, Germany, 2013

[21] L. Hannachi, O. Asfari, N. Benblidia, F. Bentayeb, N. Kabachi and O. Boussaid, "Community Extraction Based on Topic-Driven-Model for Clustering Users Tweets," in Proceedings of Advanced Data Mining and Applications, Berlin, Heidelberg, 2012.

[22] S. Jaffali, S. Jamoussi and A. B. Hamadou, "Grouping Like-Minded Users Based on Text and Sentiment Analysis," in Proceedings of Computational Collective Intelligence Technologies and Applications, 2014.

[23] J. Weng, E. Lim, J. Jiang and Q. He, "Twitterrank: finding topic-sensitive influential twitterers," in Proceedings of the third ACM international conference on Web search and data mining, 2010

[24] R. I. M. Dunbar, "Do online social media cut through the constraints that limit the size of offline social networks," Royal Society Open Science, vol. 3, no. 1, p. 150292, 2016.

[25] "Apache Lucene - Welcome to Apache Lucene," [Online]. Available: http://lucene.apache.org/. [Accessed 22 2 2019]

[26] "Index of /simplewiki/.," [Online]. Available: https://dumps.wikimedia.org/simplewiki. [Accessed 21 2 2019].

[27] S. Deerwester, "Indexing by latent semantic analysis," Journal of the Association for Information Science&Technology, vol. 41, no. 6, pp. 391-407, 2010.

[28] R. Cooley, B. Mobasher and J. Srivastava, "Web Mining:Information and Pattern Discovery on the World Web," in Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), 1997.

[29] K. Lagus and S. Kaski, "Keyword selection method for characterizing text documents maps," in Artificial Neural Networks, 1999..

[30] G. Salton and M. J. McGill, "The SMART and SIREExperimental Retrieval Systems," in Readings in Information Retrieval. Morgan Kaufmann Publishers Inc, 1997

[31] "Hakia's Semantic Search: The Answer to Poor Keyword Based Relevancy," [Online]. Available: www.searchenginejournal.com. [Accessed 15 Feb 2019].

[32] "Semantic SearchLexxe : Search Engine that Answers Exact Queries," [Online]. Available: www.searchenginejournal.com. [Accessed 13 Jan 2019].

[33] "Factbites Search Engine Encyclopedia Hybrid," [Online]. Available: www.searchenginejournal.com. [Accessed 26 Jan 2019].

[34] Summarization, the Answer to Web Search: Interview with Dmitri Soubbotin of SenseBot, [Online]. Available: www.searchenginejournal.com. [Accessed 21 Feb 2019].