Deanship of Graduate Studies

Al-Quds University

Multi-Agent Semantic Social Networks Based on Tag Rank

Sameh Abdelfattah Hussein Awad

M.Sc. Thesis

Jerusalem-Palestine

1439-2018

Multi-Agent Semantic Social Networks Based on Tag Rank

Prepared By:

Sameh Abdelfattah Hussein Awad

B.Sc. Computer Engineering, Al-Quds University, Palestine.

Supervisor: Dr. Rushdi Hamamreh

A thesis submitted to the Faculty of Engineering, Al-Quds University in Partial fulfilment of the requirements for the degree of Master of Electronic and Computer Engineering.

1439 - 2018

Al-Quds University

Deanship of Graduate Studies

Electronic and Computer Engineering

Thesis Approval

Multi-Agent Semantic Social Networks Based on Tag Rank

Prepared by: Sameh Abdelfattah Hussein Awad

Registration No. 21510019

Supervisor: Dr. Rushdi Hamamreh

Master thesis submitted and accepted. Date: 13 /5 /2018

The names and signatures of the examining committee members are as follows:

1- Head of Committee: Dr. Rushdi Hamamreh Signature:
2- Internal Examiner: Dr. Saeed Salah Signature:
3- External Examiner: Dr. Iyad Tumar Signature:

Jerusalem – Palestine

1439 – 2018

Dedication

To the memory of my father (may Allah grant him His Mercy).

To my mother who has been supporting and encouraging me all the way.

To my beloved wife, for her outstanding and highly appreciated patience day and night throughout the time of my study.

To my son and daughter.

To my brother and sister.

To the great Umma of Islam, which making its way back to its original position…. The best Umma brought to all humankind.

Sameh

Declaration

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed.....

Sameh Abdelfattah Hussein Awad

Date: 13 /5/2018

Acknowledgments

All Praise to ALLAH, the Almighty (swt), the greatest of all, on whom ultimately we depend for sustenance and guidance. I would like to thank. Almighty Allah for giving me opportunity, determination and strength to do my research.

Peace and blessing of Allah be upon the best of humankind, the Messenger of Allah, Prophet Muhammad (pbuh).

Praise, appreciation and gratitude to the great Muslim scientist Muhammad ibn Musa al-Khwarizmi, whom we use the word “Algorithm” that is derived from his name.

I would like to express my sincere gratitude and appreciation to my supervisor Dr. Rushdi Hamamreh, for the continuous support of my study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I sincerely appreciate the cooperation and help by the academic staff of the Faculty of Engineering at Al-Quds University.

I am also not forgetting my colleagues in the Department of Information Technology at Birzeit University who support and help me during my study years.

Many thanks to my Master study colleagues and everyone shared me his feelings, encouragement and support.

Abstract

Social Networks has become one of the most popular platforms to allow users to communicate, and share their interests without being at the same geographical location. The great and rapid growth of Social Media sites such as Facebook, LinkedIn, Twitter, etc. causes huge amount of user-generated content. Thus, the improvement in the information quality and integrity becomes a great challenge to all social media sites, which allows users to get the desired content or be linked to the best link relation using improved search / link technique. So introducing semantics to media networks will widen up the representation of the social networks.

Semantic Social Networks representation of social links will be extended by the semantic relationships found in the vocabularies which are known as (tags) in most of social media networks.

Semantic Social Networks contents can be linked using autonomous agents, which perform specific tasks to make the linking process automated, self-learning and intelligent. Multi-agent systems concept also introduced to this approach.

In this thesis, we proposed a model of semantic social networks from the perspective of multi-agent systems (MSSNT). In this model, the multi-agent system is composed of two main functionalities: semantic indexing and tag ranking.

The proposed model is an improvement of the output of ranking, and to achieve that some kind of filters should be used to increases the rank of content. And improving the rank must be met by semantic content analysis that makes the linking similar according to subjects or keywords on the social media content.

The proposed model for the social media engine is based on Enhanced Latent Dirichlet Allocation (E-LDA) as a semantic indexing algorithm, combined with Tag Rank as social network ranking algorithm.

Simulation Results have shown better performance in both indexing and ranking phases. In indexing phase, E-LDA algorithm produces better precision and recall with 4% than LDA basic algorithm, and absolutely best performance comparing with the previous indexing algorithms used in web.

In ranking phase, Tag Rank algorithm based on topic per document distribution resulting from E-LDA have shown better performance in precision and recall with approximately 5%. And best results in Mean Average Precision(MAP) and Normalized Discounted Cumulative Gain (NDCG) comparing to other ranking algorithms.

Table of Contents

List of Figures

List of Algorithms

List of Tables

Chapter One: Introduction

Chapter One:

Introduction:

1.1 Introduction

Social media are emerging field in information interchange, worldwide used and wanted. It is a challenging subject to do a research in social media field as it was and still affecting us in every aspect of our lives [1].

Ellison and Boyd defined social networks (SN) as web-based services that allow users to build a public or semi-public profile within a system, connect to a list of other users by sharing a connection, and view and extend their list of connections and those made by others within the system. The nature of these connections may vary from (SN) site to another [2].

While WonKim et al. had defined the social Websites as those Websites that facilitate the formation of online communities to the people, and share user-created contents (UCCs). The people may restricted users who are members of a closed community or organization, such as universities, corporations, political societies and parties, or professional societies. On the other hand, people may be users of the open [3].

The society may be a network of friends whose friendship is extended to be online, online colleagues, or interest groups based on hobbies, interests, causes, professions, ethnicity, or gender, etc.). The UCC maybe pictures, videos, bookmarks of web links, users' profiles, user's activities, text such as blog, micro blog, and comments, etc.

The sharing of the UCC means posting, watching, and commenting of the UCC, also it may include voting, saving, and re-broadcast the UCC.

The improvement in retrieved contents in social media should be given attention as it reflects the quality and integrity of social media in general. The new perspective was to introduce semantics into social network to get Semantic Social Network (SSN) in which relations and social graph are built according to the words meanings, especially keywords which are widely-known as Tags in the social media network.

In current social media networks, links between contents are constructed by many ranking techniques according to the way to deal with data, importance and priority of data. Such as posts on Facebook, hashtags in Twitter, and jobs and experiences in LinkedIn, etc. And so data must be ranked in a way that links constructing the social graph will reflect natural distribution and connection between nodes of the social media. A rank for each node is given by making iterative process of weights in network. In Semantic Social Networks, this weight can be given according to semantic content of the social media node.

1.2 Overview of Semantic Social Networks

Semantic Content of Semantic Social Network which is large and complex collections of data and that is known nowadays as "Big Data"[4] must be indexed before ranking process. This can be achieved by introducing semantic indexing algorithms to process content of Semantic Social Networks[5].

Improving indexing output and choosing the proper rank algorithm will affect the quality of the social graph and how nodes will be linked in semantic social network.

The existence of various ranking algorithms depending on how dealing with content which affects the quality of the output of the ranking. So the ranking of contents in social media should be based on some criteria that reflects really-related topics or links to the content. This can be achieved by depending on semantic indexing algorithms that gives the actual relations depending on the topic of the contents. Figure 1.1 shows an example of semantic social network.

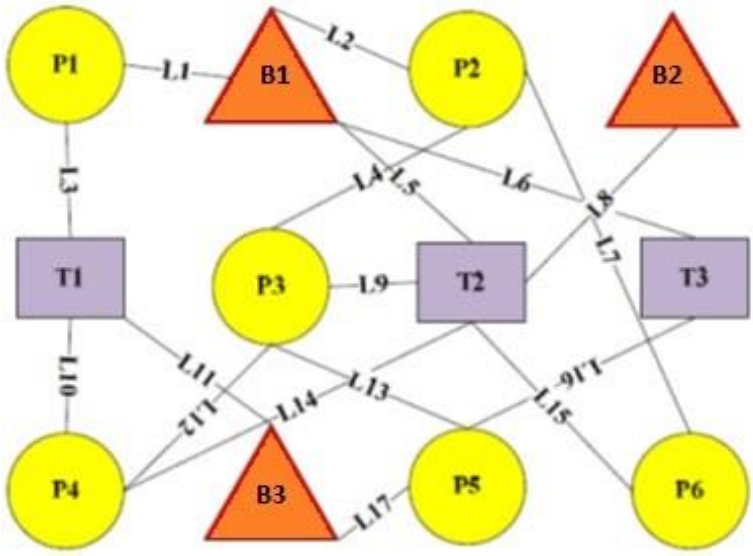


Figure 1.1 An Example of Semantic Social Network [6]

The circles represent people (such as P1), and the squares represent things (contents) in the semantic social network (such as T1), and triangles represent behavior of people and things (such as B1). The lines between these entities represent the semantic social link between people, things and behaviors [6].

1.3 Overview of Multi-Agent Systems

For Indexing and Ranking processes. The Concept of Multi-Agent system (MAS) is a great addition to give good, improving, and self-learning mechanism especially in social networks. Multi-Agent Systems are computerized system consisted of multiple agents that interact intelligently within the environment which can be used to solve problems [7].

An agent is a computerized system that is qualified to do actions independently on behalf of the user or the owner by determining what is needed to be done to achieve the objectives that it was designed to achieve, rather than just carrying out the actions that it was being programmed to do directly [8].

Agents have different categories starting from simple agents to complex ones. Some categorizations suggested categorizing agents to passive agents, which means agents with no goals like obstacle, or key in any simple simulation, and active agents, which means agents implemented with simple goals like birds in flocking, and cognitive agents, which can carry out complex computations and self-learning abilities [8]. In addition, agent environments can be divided

into three main categories: virtual environment, discrete environment and continuous environment.

Agent environments have many properties, such as accessibility, which means that the possibility of the agent to gather complete information about the environment, determinism which focused on the probability that a definite effect can be caused by an action performed in the environment, periodicity which shows if an agent actions in certain time could affect other periods) [9], and dimensionality which checks if the spatial characteristics are important factors in the environment and if the agent considers space in its decision making process [10]. Agent actions in an environment are interfered using a specific middleware. Moreover, this middleware offers a design that reflects the concept of multi-agent systems, providing ways to manage resource access and making the needed agent coordination [11].

The agents in a multi-agent system have several important characteristics [7]:

- **Autonomy**: which means that agents are independent, self-aware, and autonomous.
- **Local views:** it means that there is no agent that has a full vision of the system, or the system is complicated that an agent cannot make practical utilization of such knowledge.
- **Decentralization:** which means that there is no agent that is acting as central agent controlling the other agents [12].

Figure 1.2 shows a simple concept of an intelligent agent job and interactions [13].

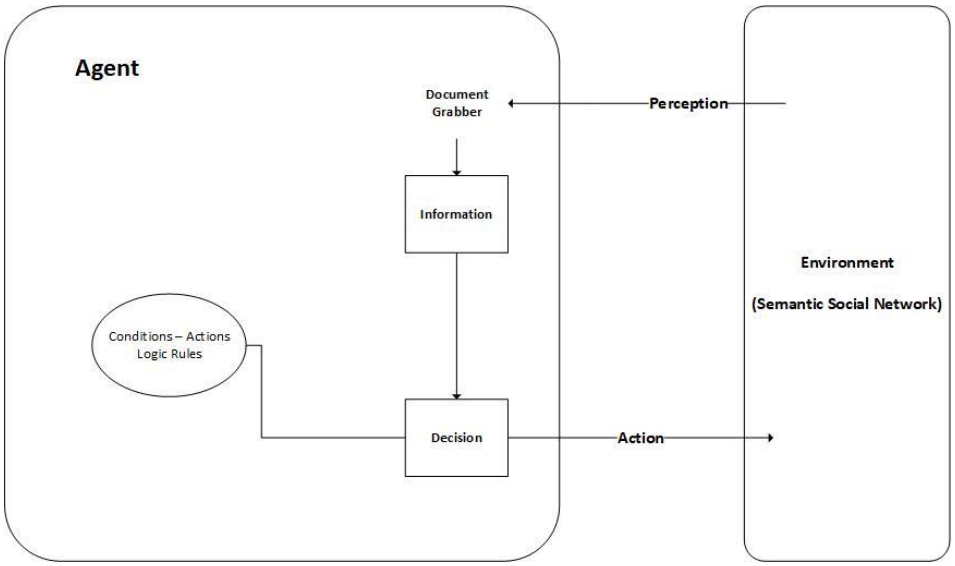


Figure 1.2 An Example of Simple Agent Process [13]

As shown in Figure 1.2, the agent role is to percept data from the environment which is the semantic social network and grabs the documents and based on programmed logic rules the agent make the decision and do the needed action in the environment. This type of agents makes the decision as a reaction to specific changes in the environment. And for that it is defined as simple ones. However, there is more intelligent type of agents which is the self-learning agent, which makes the decision not only based on logic rules that are programmed in the agent but also based on learning information that are stored from previous experiments. Figure 1.3 shows the concept of self-learning intelligent agent [13].

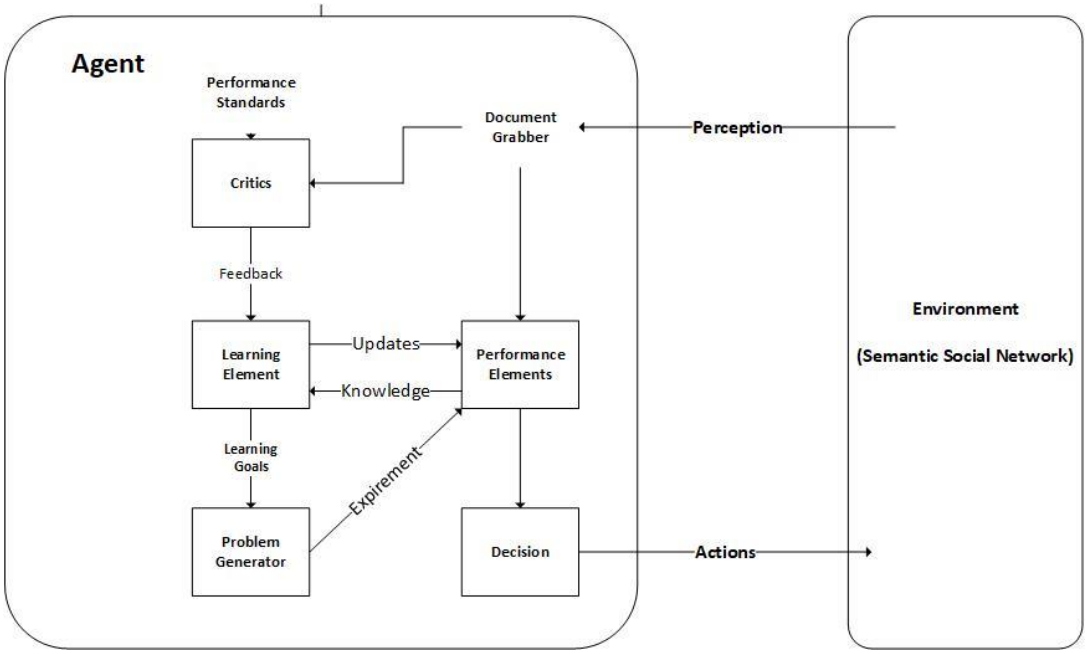


Figure 1.3 An Example of Self-Learning Intelligent Agent [13]

As shown in Figure 1.3, the agent percepts data and grabs the documents from the environment (SSN) and by using the learning elements it updates the performance elements. The agent builds the knowledge base by updating the learning elements based on critics that represent the feedback from the whole system that the agent is working in. And this knowledge base generates problems that can be used as condition rules to be used in the decision that the agent will make to do the needed action in the environment.

In social networks, the multi-agent implementation theories have two main perspectives: user perspective and network perspective.

In user perspective, the agents will be the user accounts [20], which means that each account will act as an agent in mediating data and negotiating connections with the other agents to enlarge their social network.

Nevertheless, in the network perspective, the agents will be carrying out some central operations such as filtering data, managing connectivity, and building the social graph.

Because we are discussing the semantic social network the perspective of semantics must be an important role to be done by agents in the multi-agent implementation of semantic social network.

In this thesis, we will concentrate on some roles done by agents, which are parsing data, building semantic index of the data, then ranking this index, and finally building connections between contents according to the rank output.

1.4 Research Methodology

In the thesis, the research is mainly focused on the proposing new model of Semantic Social Network based on Multiagent system. Based on our finding from the state of art and studying many researches that proposed models for SSN, we propose a model which is based on combining the proper semantic indexing algorithm with the proper ranking algorithm. Therefore, we study previous researches of the ranking algorithms used in social networks. Based on this previous work, we make the comparison between these algorithms based on the literature review in this field, and the simulation results to choose the suitable algorithm that we will use in this thesis which is the Tag Rank algorithm.

Then, we study the semantic indexing algorithms and check the strengths and weaknesses of these algorithms based on previous researches that discussed the semantic indexing. The previous researches made clear simulation results that we can conclude the proper semantic indexing algorithm to be used in this thesis. Latent Dirichlet Allocation(LDA) was the chosen indexing algorithm because the previous works have shown that it has better performance.

Then, we propose the model of Semantic Social Network based on LDA and Tag Rank algorithms. With the perspective of multi-agent systems concept.

In this thesis, we will use MATLAB as the simulation software as it is a very good in processing matrices which the data we are dealing with is matrices of semantics and indexes of these semantics.

The simulation phase is composed of two main parts, the first part is the enhancements on indexing algorithm (LDA) and comparing our work with the other algorithms based on specific metrics that will be discussed in section 1.7 in this chapter. This comparison shows that the proposed enhancement will get better index results.

The second part is combining the indexing with ranking by getting the index from LDA and rank this index by Tag Rank and comparing our results with

previous work. Showing by using specific metrics for ranking that our algorithm produces better results.

1.5 Motivation

Social media are emerging field in information interchange, worldwide used and wanted. Social medial also affects many aspects of our day life.

The social media networks facing problem due to fast data increasing and the amount of data is getting big in high rates. This data in most of the social media networks does not reflect true or actual reality. On the contrary, the integrity of data depends on user's inputs.

The data in social media network should be parsed, indexed and ranked according to the topic of the content entered by the user, which means that data in social media network must be processed by semantic indexing algorithms and then ranked by social ranking algorithm according to the index result. The improvement of the retrieved contents in social media on the criteria above will give us quality and integrity of data in social media networks.

Moreover, the processes of semantic indexing and ranking should be autonomous, and here the idea of multi-agent systems can achieve this goal.

This motivates us to propose new perspective of social media networks, which is the multi-agent semantic social network. In this proposed perspective, we connect the semantic index with the social ranking algorithms. In addition, each algorithm will be the role of an agent that carries out this task in the proposed model.

The proposed model shows an enhanced semantic indexing algorithm. And also ranking algorithm which reflects more integrity in social media network.

1.6 Problem Statement

The semantic social networks have many different ranking algorithms depending on how contents are dealt with. Most of indexing algorithms are using term-frequency-based indexing. And if we need to improve the quality of the output of the ranking we should use good criteria which reflects the integrity of the contents.

There are many semantic indexing algorithms that are used. So we need to choose the proper algorithm that will produce enhancement in precision and recall for indexing output.

In SSN, ranking algorithms used different parameters to rank the content that are based on users' input. So using rank algorithm that reflects the actual content of SN.

1.7 Metrics for Evaluating Indexing and Ranking

In this thesis, we use main four metrics; two for evaluating indexing which are precision and recall. The other two is for evaluating ranking and these metrics are mean average precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) [24].

1) **Precision:** is the relation between the number of relevant documents retrieved to the total number of irrelevant and relevant documents retrieved, expressed usually as a percentage.
2) **Recall:** is the relation between the number of relevant documents retrieved to the total number of relevant documents in the dataset, expressed as a percentage. Figure 1.4 shows the relation between precision and recall.

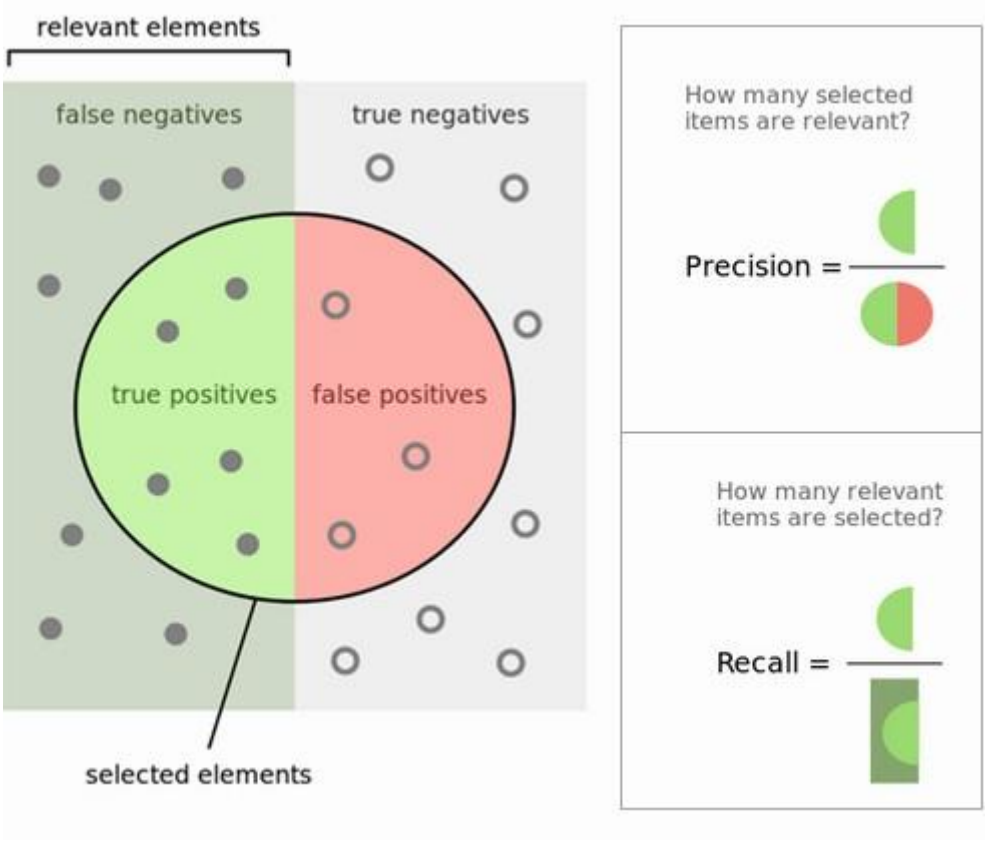


Figure 1.4 Precision and Recall [24]

3) **Mean Average Precision (MAP):** is the precision-at-*k* score of a ranking *y*, averaged over all the positions *k* of relevant documents.
4) **Normalized Discounted Cumulative Gain (NDCG):** is a normalization of the Discounted Cumulative Gain (DCG) where (DCG) is a weighted sum of the relevancy degree of the ranked items. The weight is a decreasing function of the rank of the object, and therefore called discount. The reason for using the discount is that the probability that a user views a document decreases with respect to its rank.

In chapter 5, in the simulation results discussion we will discuss these metrics and show their mathematical equations to evaluate our results.

1.8 Thesis Contributions

As data integrity is one of the main challenges in semantic social networks, many methods were introduced to improve the quality of information depending on the way this information processed. So the proper indexing and

ranking the data in semantic social network will definitely give its impact on the results.

In this thesis, we have done the following contributions:

- We have Proposed a new novel model of semantic social network based on the concept of multi-agent systems.
- We have designed new architecture of (SSN) based on combining semantic indexing algorithm (LDA) with Tag Rank.
- We have enhanced (LDA) in a new improved algorithm (E-LDA) because we have chosen the proper parameters of the algorithm to get the best output.
- Enhanced- LDA (E-LDA) is also improved by designing a filter to get the best output from semantic indexing algorithm to be passed to the ranking algorithm.
- Enhanced- LDA (E-LDA) combined with Tag Rank are improved algorithms because the data processed reflects the self-learning feature which can be applied to multi-agent systems.

1.9 Literature Review

In this literature review, some research studies concentrated on the application of multi-agent systems in social networks and also some outreach for semantic social networks. While other researches discussed the implementation of several ranking techniques in social networks. While some other papers tried to apply the semantic indexing techniques in social networks.

Vadoodparast and Taghiyareh proposed a multi-agent structure called MAFIM, which will be used to maximize the use of the product in dynamic social networks. MAFIM consists of two types of agents: model agents and solution provider agents. These agents view the dynamic social network as successive static network snapshots and, in connection with this, choose a budget assignment policy so that each snapshot gets its share from the budget determined by the sales manager. Based on MAFIM, the authors presented MASPEL - a single product model that takes into account network communities, their judgments about each other and their ability to profit. MASPEL uses a specific budget assignment policy, in which budgets are assigned to advertising campaigns in a gradually decreasing manner. This

study showed that it is more effective to run many short-lived campaigns instead of a few long-lived ones [14].

Wang and Djurić suggested that agents supposed to build knowledge from the decisions of previous agents and update this "beliefs" using Bayesian theory. The authors defined the concept of social belief in the truthfulness of the two hypotheses and gave results on the convergence of social beliefs. They also proved that with the proposed random policy it is possible to avoid the information cascade and to obtain asymptotic training. They then applied a random policy to data models that represent observations on the distribution belonging to an exponential family [15].

While Zhang, et al. described the conceptual structure of semantics in the cloud. The structure consists of 8 parts. Their model includes: the semantic interface of social search, the semantic parser, the semantic social rank, the semantic index base, the base of social relations, the module of semantic social computing, the massive processing of data and the distributed file system "Hadoop" [6].

Jiang, et al. presented a new model of task distribution based on the mechanism of relational reputation, where the agent's past behavior in matching the resources of the task can affect its probability of distributing new tasks in the future. In this model, an agent who introduces more reliable resources with less access time during the execution of the task gets a higher reputation in the negotiations and can receive a discriminatory distribution of new tasks [16].

In the field of Tag ranking, Montañés, et al. offered a tag recommendation based on logistic regression, which, according to their research, is free from the use of content information, providing ranking of certain tags and learning only from relationships among users, previously placed resources and tags, avoiding the cost of using the content of resources [17].

Lu, et al. proposed a system of social re-registration to search for images based on tags, taking into account the relevance and diversity of the image. They are aimed at reorienting images in accordance with their visual information, semantic information and social prompts. Initial results include images contributed by various social users. Usually, each user contributes several images. First, sort these images by re-ranking between users. Users who have a higher contribution to this query are higher. Then, the user is re-ranked sequentially within the set of images of ranked users, and only the most relevant image from the set of images of each user is selected. These

selected images are the final results. The authors also built an inverted index structure for the social image data set to speed up the search process [18].
While Qian, et al. suggested an approach to re-tag social images with diverse semantics. Both the relevance of the tag to the image and its semantic compensation for already defined tags are merged to determine the final tag list for the image. Unlike existing approaches to image placement, top-level ranked tags are not only very important to the image, but also have significant semantic compensation to each other [19].
For the implementation of MAS in social networks Enrico Franchi introduced a multi-agent system that implements a fully distributed social networking system that supports user profiles as Friend-Of-A-Friend (FOAF) profiles. This means that users should be the sole owner of the information they provide and solve design privacy issues. And users are represented by agents who provide mediation in access to confidential data and actively interact with other agents to expand their social network to their users. The author introduced the distributed connection detection algorithm used by agents, and detailed the presentation of data in the users' profiles used to support the algorithm [20].
The approach of the Fengs and Jin was to rank the tags in descending order of their correspondence to this image and that, according to the authors, greatly simplifies the problem. In addition, the authors proposed a method that combines prediction models for different tags in a matrix and discards the rating of tags in the problem of matrix reconstruction. What introduces matrix tracing is to explicitly control the complexity of the model, so a reliable prediction model can be learned for tag ranking even if the tag space is large and the number of training images is limited [21].

Zhang, et al. proposed a personalized method for estimating social images based on a custom image model. The purpose of this is the effective use of tags, the social image tags are redistributed according to the contents of the image and to obtain user preferences, the custom image-tag model is constructed with a three-way graph in accordance with the relationship between users, images and vertices-volume tags; and a personal system of social recommendations is implemented on the basis of the user image model [22].

As shown in this literature review there are many suggested models to represent semantic social network depending on the concept of multi agent systems. Also other models discussed suggestions how to implement tags in

semantic social network. While other researches concentrated on the improvement of semantics structure in social network.

All of the researches discussed in this section guided us in our research track to represent the multi agent implementation of semantic social network based on semantic indexing and tag ranking. The next section shows the structure of our research.

1.10 Thesis Outline

The rest of the thesis is organized as follows:

This thesis proposes a new model of semantic social network based on Tag Rank. Hence, in chapter 2 we discuss the ranking algorithms. Including discussing the classification depending on the web mining techniques used. Showing the mathematical models of the ranking algorithms. Comparing their parameters, importance, and limitations. And summarizing with showing the decision of what algorithm is chosen in this thesis.

Chapter 3 is about semantic indexing algorithms. It shows classification of these algorithms and the criteria of the indexing process. Discussing the models of these indexing algorithms, their strengths and weaknesses. Comparing them based on previous researches. Summarize the chapter with the decision of the chosen algorithm in this thesis.

In chapter 4, we represent the proposed model. Which is a new architecture of the social media engine based on the concept of multi-agent systems. Discussing the behavior of this model by flowcharts, pseudocode and AUML sequence diagram. Also representing the mathematical model and the functions to be used in the simulation.

Chapter 5 is about simulation and results. The start is with introducing the used simulation tools in this research and a description of the dataset used in simulation. Then the metrics that were considered in this research are provided. After that, the results of the simulation are provided with the analysis of these results.

Chapter 6 provides the conclusion of this thesis and suggests future work to improve and enhance the field of this research.

Chapter Two: Ranking in Social Networks

Chapter Two:

Ranking in Social Networks:

2.1 Introduction

Data mining is the extraction or extraction of knowledge from many data called Knowledge Discovery in Databases (KDD) [23], which is the result of the natural evolution of information technology. The technology of data mining attaches great importance to the development of information industries, which are developing rapidly. The intellectual processing of data mining is influenced by numerous practices, including database system, statistics, machine learning and data analysis, etc. Many methods of data mining and special tools are available nowadays. In many areas of research, data was used, such as a database, data analysis and machine learning. Information Retrieval (IR) is a method used in Data Mining for searching in huge databases for obtaining related documents. IR refers to the science of information retrieval in documents, texts, relational databases, multimedia files and on the World Wide Web (WWW). Many users are concerned about the area (IR), such as professional researchers, librarians, strategic and political analysts and marketers. Applications of (IR) are different and not only exclude extracting information from large documents, filtering spam, probing in digital libraries, filtering information, extracting objects from images, classifying and clustering documents and searching the Internet. With the increase in the number of web pages and users on the Internet [25], the number of requests sent to search engines is also growing rapidly. Therefore, search engines should be more efficient in their way of their processing and their output. Web mining techniques are introduced in the search engines to extract relevant documents from the Web database and provide the necessary information to the manipulators.

Search engines have become prosperous and trendy if they use the proper ranking methods, and these days it is very thriving because of the use of the ranking algorithm.

Search engines use ranking algorithms to view search results based on reputation, relevance, and results of content and web mining techniques to order them in the same way as the user [26]. Some ranking algorithms depend only on the structure of document links, which means their popularity, and they are called algorithms for intellectual analysis of the web structure mining algorithms. Other algorithms focus on content in documents that are known as web content mining algorithms, and some use a modification of the content of the document, as well as a link structure to determine the result of a rank for a particular document. If the displayed search results are not displayed in accordance with the user's interest, the search engine will lose its reputation. Thus, ranking algorithms have become very important for search engines [27].

2.2 Web Mining Overview

Web mining is the mechanism for classifying web pages and Internet users, taking into account the content of the page and the behavior of the Internet user in the past. The application of data mining technology is web development, which is used spontaneously to search and retrieve information from the World Wide Web (WWW). In accordance with the purposes of the analysis, Web Mining has three main branches, which are: Web Content Mining (WCM), Web Structure Mining (WSM) and Web Usage Mining (WUM) [28].

WCM is the process of extracting useful information from the content of web credentials. Web-based credentials can consist of text, audio, video, images, or structured records, such as tables and lists. Mining can be targeted on web documents, as well as on pages of results created with the help of a search engine. There are two approaches in the field of content mining, called the agent-based approach, and the database-based approach. The method of the agent is based on the search for suitable information using the uniqueness of a specific domain for the interpretation and organization of the information collected. The database approach is used to return semi-structure data from the Internet.

WUM is the method of extracting useful information from secondary data resulting from user interactions during surfing on the Internet. It retrieves data collected in server referrer logs, access logs, agent logs, user profiles, and cookies on the metadata side.

The aim of the WSM is to create a structural abstraction about a website and a web page. He tries to determine the structure of hyperlink links at the level of the document "bury". Based on the hyperlink topology, intelligent processing of the web structure will classify web pages and generate information similar to the similarity and interconnection between different websites.

This type of mining can be performed at the document level (intra-page) or at the hyperlink level (inter-page). It is important to evaluate the structure of web data for information retrieval. Three categories of web development and own application areas are described, including site improvement, business analytics, web personalization, site modification, usage characteristics and page ranking, classification etc. Figure 2.1 [29] shows an overview of the classification of the web mining types.

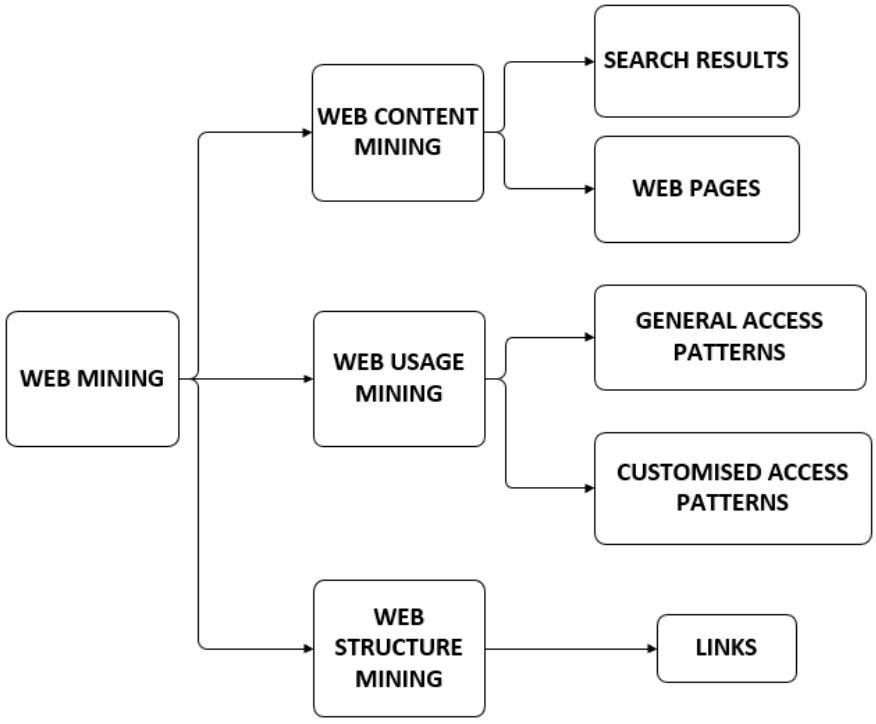


Figure 2.1 Classification of Web Mining Types [29]

2.3 Ranking Algorithms

2.3.1 Page Rank:

Usually in Google, in Page Rank, if the page contains important links to it, links to this page on another page should also be considered important pages. And PageRank finds feedback in determining the score of the rank.

PageRank works by counting the number and quality of links to a page to determine an approximate estimate of how important a website is. The initial assumption is that more important websites are likely to get more links from other sites. The rank of the page considers the feedback in determining the score of the rank [30].

So assume we have two pages, *u* and *v*:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)} \qquad (2.1)$$

Where B_u is the set of all pages linked to page u, and $L(v)$ is the number of links from page v.

Considering the damping factor, the page rank will be:

$$PR(u) = (1 - d) + d \sum_{v \in B_u} \frac{PR(v)}{L(v)} \qquad (2.2)$$

This ranking algorithm does not reflect the content of the pages, but it concentrates on the number of links associated with the page. Figure 2.2 shows an illustration of page rank algorithm.

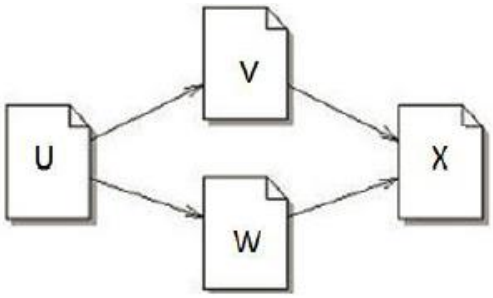


Figure 2.2 Page Rank Algorithm Illustration [30]

2.3.2 Weighted Page Rank:

Weighted Page Rank algorithm (WPR) is an improvement of the original PageRank ranking algorithm [31]. (WPR) decides ranking score based on the popularity of pages, taking into account the importance of both page in-links and page out-links. This algorithm provides a high rank value for more popular pages and does not divide the rank of the page among its link pages. Each page of links is assigned a rating value based on its popularity. The popularity of the page is determined by monitoring the number of incoming and outgoing links. For example, for the pages u, p and v the weight rank is:

$$W_{(v,u)}^{in} = \frac{I_u}{\sum_{p \in R(v)} I_p} \tag{2.3}$$

Where I_u, I_p are numbers of in-links of pages u and p, $R(v)$ is the reference page list of page v

$$W_{(v,u)}^{out} = \frac{O_u}{\sum_{p \in R(v)} O_p} \tag{2.4}$$

Where O_u, O_p are numbers of out-links of pages u and p.

2.3.3 Hyper-link Induced Topic Search (HITS) Algorithm:

The HITS algorithm evaluates the web page by processing links and references. In HITS, a web page is called authority and a hub, if the web page is indicated by many hyperlinks, it is called "Power", and if the page is pointing to different hyperlinks and the web page is called a hub [32]. Concentrators are pages that act as resource lists. Authorities are pages containing the main content. A decent hub is a page that points to many authoritative pages of this content, and a good authority is the page indicated by many good hub pages on the same content. A page can be a good center and a good authority at the same time. He uses an iterative algorithm to calculate the concentrator and authoritative weights.

The HITS algorithm consists of two steps: the first step is the sampling step, the second is the iterative step. Sampling step is a set of relevant pages for this query is gathered. While in the iterative step, Hubs and authorities are detected using the output of the sampling step. Following expressions are used to calculate the weight of Hub (*Hp*) and the weight of Authority (*Ap*).

$$HUB\ (H_P) = \sum_{q \in I_P} A_q \tag{2.5}$$

$$AUTHORITY\ (A_P) = \sum_{q \in B_P} H_q \tag{2.6}$$

Here Hub score of a page is (*Hq*) and authority score of page is (*Aq*). *I(p)* is set of reference pages of page *p* and *B*(*p*) is set of referrer pages of page *p*. The weight of the authority pages is proportional to the weights of the hub pages that reference to the authority page. While another one is, the weight of the hub page is proportional to the weights of the authority pages that the hub pages' links. Figure 2.3 shows Hubs and Authorities in HITS rank algorithm.

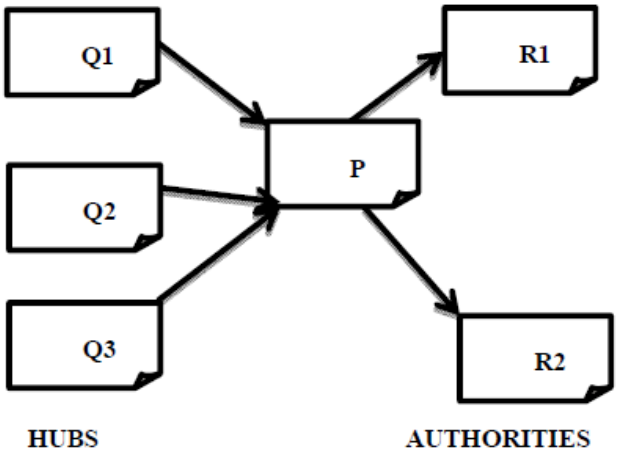


Figure 2.3 HITS Hubs and Authorities [32]

As shown on Figure2.3. The calculation of Hubs and Authorities will be:

$$A_P = H_{Q1} + H_{Q2} + H_{Q3} \tag{2.7}$$

$$H_P = A_{R1} + A_{R2} \tag{2.8}$$

2.3.4 Time Rank Algorithm:

The Time Rank algorithm is used to improve rank score by using the time of visiting a web page [33]. This algorithm measures the time it takes to visit the page after applying the original and improved methods of the ranking algorithm of the web page to find out about the degree of importance for users. This algorithm uses the time factor to improve the accuracy of the ranking of a web page. Time rank can be assumed a web usage mining algorithm structure. The mathematical model of this algorithm is as following:

$$P_r\big(T(i)|_q\big) = P_r\big(T(i)\big) * P_r\big(q|T(i)\big) \tag{2.9}$$

Where $T(i)$ is the topic *i* of each page. And $P_r(T(i))$means the proportion of pages related to topic I in the whole pages set. $P_r(T(i)|_q)$ Means the probability of the query q belonging to topic *i*.

The topic sensitive page rank used in the Time Rank is given by:

$$TSPR\ t(i) = a\sum_{i\in B}\frac{TSPR(i)}{|F_i|} + (1-a).Et(i) \qquad (2.10)$$

Where single jump probability 1/n is replaced by Et = {E(1), E(2).....E(n)}, n is the no. of topics.

$$E(i) = \begin{cases} 1/nt \\ 0 \end{cases} \qquad (2.11)$$

Where nt number of pages related to topic.

There are n TSPR scores corresponding to the topics. It is calculated statically offline. After some time of search engines running, the time vector associated with the topics for each page can be calculated, and therefore, each page is assigned as a page rank depending on the time of visit.

$$TIME\ PR_t(j) = TSPR_t(j) * T(t) \qquad (2.12)$$

Where time vector Tv = {T(1), T(2), ...T(n)}. And T(i) is the user's total visiting time of a page related to topic i.

Time Rank means that regardless of the similarity of the similar structure of the links of the two web pages, the page with a longer visit time gets a high score.

2.3.5 Edge Rank:

Edge Rank is this name an algorithm that was used by Facebook to determine which articles should be displayed in the user's news feed [34]. Every action that their friends take is potential newsfeed story. Facebook calls each one of these actions as an "Edge". This means that every time a friend sends a status update, comments on another status update, tags a photo, joins a fan page or an RSVP response (a response from an invited person) to the event that it generates an "Edge", and the story of this Edge can appear in the user's private newsfeed.

It would be absolutely amazing if all the possible stories of your friends were shown in the news line. Therefore, Facebook created an algorithm to predict how interesting each story is for each user. Facebook calls this algorithm "Edge Rank", because it ranks the edge. Then they filter each user's news feed to show only the most popular stories for that particular user . The general equation of this algorithm is:

$$Edge\ Rank = u_e * w_e * d_e \tag{2.13}$$

Where u_e is the affinity score (between viewing users and edge creator). w_e Weight for the edge type (create, comment, like, tag, etc.) and d_e time decay factor.

2.3.6 Tag Rank:

Tag Rank is a new suggested technique similar to the page rank, but it works with tags and links between nodes, depending on the presence of a tag in the content of social networks [35].

This algorithm digs out the behavior of web user annotations, calculates the heat of the tags. Using the time factor of the new data source tag and the behavior of Web user annotations. It can respond to the true quality of tags more externally and improve the reliability of page ranking. This algorithm provides the best authentication method for ranking web pages. The results of this algorithm are very accurate, and this algorithm better indexes new information resources.

In a simple way, when we get the semantic index by the indexing agent, the task of the rating agent is to build a weight matrix of Tag-Pair (TWM) as a rank matrix depending on the result of indexing. The mathematical model of the Tag Rank based on the research of DaeHoon Hwang can be summarized as following [36] [37].:

First creating TFM which is (Tag Frequency Matrix) which is the sum of Tag Matrices TM depending on tag simultaneous appearance:

So $TM_{(i,j)} = 0: tag\ i\ and\ tag\ j$ do not appear simultaneously on certain content.

And $TM_{(i,j)} = 1: tag\ i\ and\ tag\ j$ appear simultaneously on certain content.

And so Tag Frequency Matrix is

$$TFM_{(i,j)} = \sum_{k=1}^{m} TM_k(i,j) \tag{2.14}$$

Lastly, the Tag-Pair Weight Matrix can be computed as follows:

$$TWM_{(i,j)} = TSM_{(i,j)} \times TFM_{(i,j)} \tag{2.15}$$

Where $TSM_{(i,j)}$ is an entry of tag-pair similarity matrix.

Tag Rank algorithm is so vital in social network as it uses tags which are very important content in most of social media networks.

2.4 Comparison of Ranking Algorithms

Table 2.1 shows a comparison between the ranking algorithms discussed in this chapter [38] [39]:

Table 2.1 A Comparison between Rank algorithms

Algorithm	***Web Mining Technique***	***Methodology***	***Input Parameters***	***Importance***	***Limitations***
Page Rank	Web Structure Mining	Computing the score for pages at the time of indexing them.	Back Links	Used in websites which receive links from other websites.	Results come at indexing time, not on query time
Weighted Page Rank	Web Structure Mining	Calculating weight of web pages on the basis of weight and importance of input and outgoing links	Back Links and Forward Links	Considering popularity of the page beside the in and out links.	Ignoring relevancy.
HITS	Web Structure/ Content Mining	Computing hubs and authority of the relevant pages.	Content, Back Links and Forward Links.	Concentrates on the relevancy and referencing of Hub and Authority pages.	Topic drift and efficiency problems
Time Rank	Web Usage Mining	Using visiting time to be added to the computational score of page rank algorithm of the page.	Original Page Rank and Server Log.	Improves the page rank score by adding the visiting time.	Important pages are ignored because they increase the rank of web pages, which are opened for long time.
Edge Rank	Web Content Mining	Computing Affinity score and edge type product	Affinity Score/ Edge Type	Good for Social network.	Promoting content affects score.
Tag Rank	Web Content Mining	Sequential clicking for sequence vector calculation with the use of random surfing model.	Popular Tags/ related Bookmarks .	High for social networks.	The user inserts the tag and may be irrelevant.

Table 2.1 shows the web mining technique used for each ranking algorithms. Also it explains the methodology of each algorithm, its input parameters, its importance by showing the strength points, and the limitation, which is the weaknesses of each algorithm. As shown in Table 2.1, the most suitable input parameter for our research is the keywords, tags and bookmarks.

Which means that Tag Rank is the best ranking algorithm based on the input parameter. While PR, WPR and HITS concentrate on the back links and forward links. In addition, Time Rank concentrates on the visiting time, which can be obtained from the server log. Moreover, Edge Rank gets the Affinity score and edge type as input parameters.

As shown in Table 2.1, Tag Rank is also important for social networks, and this means that in semantic social networks Tag Rank can be more reliable

ranking algorithm. On the other hand, Edge Rank was good algorithm for social networks but it is not good for semantic social networks because it did not deal with the semantic contents of the edges. PR, WPR and HITS deal with pages, their links, their popularity, their relevancy and referencing and that is not reflecting the actual perspective of the semantic social network. Time Rank also adds the visiting time to the page rank score.

PR has drawback that the results come at the indexing time not on the query time, which means any updates after indexing are not included in the time of retrieval of the information. WPR drawback is ignoring the relevancy and that is because it depends on the popularity caused by out links. HITS has problems in efficiency and topic drifting issues that affects the output to be irrelevant to search or link topic. Time Rank sometimes ignores important pages because they were opened for a long time and by its algorithm will be dropped. Edge Rank problem is that the promoted contents is affecting the rank score and that means ranking is not only based on data but also in marketing this data to be shown to the user on the personal news feeds. Tag Rank problem is that the user selects the tags of the content and that may cause irrelevant links to be created based on this created tags.

2.5 Summary

As shown in Table 2.1, the Tag rank algorithm is the best algorithm to be used on semantic social network, although it has a weak point which is that tags are entered by the users.

In this thesis, we will overcome this weak point as the tags will not be entered by the users, but will be imported by semantic indexing results, which we will discuss its algorithms in the next chapter (Chapter 3). And the indexing process will be carried out by the indexing agent which will be discussed on the proposed model discussed on Chapter 4.

Chapter Three: Indexing Algorithm

3.1 Introduction

3.2 Categorization of indexing algorithms

3.3 Indexing Algorithms

3.3.1 TF-IDF

3.3.2 VSM

3.3.3 LSI

3.3.4 PLSI

3.3.5 LDA

3.4 Comparison of Indexing Algorithms

3.5 Summary

Chapter Three:

Indexing Algorithm:

3.1 Introduction

Indexing algorithms - mainly in search engines - collect, parse, analyze and store data to facilitate quick and accurate information retrieval [40]. Index design includes interdisciplinary concepts from linguistics, cognitive psychology, mathematics, computer science and informatics. An alternative name for the process in the context of search engines intended for searching web pages on the Internet is web indexing.

When dealing with information retrieval, stored documents are identified by sets of terms that are used to represent the contents of the document. The indexing process is the assignment of the index for documents in the collection of documents. The index of terms can be predefined as a fixed set of controlled vocabulary or can be any additional words that the indices consider to be related to the topic of the document.

As more and more texts are available, the indexing of the natural language and the computer choice of indexing terms from texts are becoming more and more used.

Popular search engines focus on full-text indexing of online documents in natural language. Media types, such as video, audio and graphics, are also searchable. And search engines get the best result of the semantic indexing algorithms introduced to get the actual content index of social network content.

3.2 Categorization of Indexing Algorithms

Indexing algorithms are computer science related techniques that involves machine learning. The indexing is simply building structures that approximate concepts from a large set of documents. And can be applied in construction of text mining in information retrieval systems.

Semantic Indexing Algorithms have two main generations of indexing

- Lexical-based indexing: which means that index will be constructed based on frequency of words in document and similarity between documents based on the frequency of certain word. This generation algorithms do not capture the position of word in the text, neither semantics, nor co-occurrences in different documents. In other words, these algorithms deal with bag-of words models without considering meaning. An example of these algorithms is TF-IDF, and VSM.
- Semantic- based indexing: which involves analysis of a corpus, which is the task of building structures that approximate concepts from a large set of documents. Some of this generation techniques uses natural relations between set of documents and terms they contain. Such as LSI. Others introduced probability modeling to downsize the occurrence indexes. Such as PLSI. While some other algorithms involve attributing document terms to topics. Such as LDA. But all these algorithms are concentrating on the semantic not the lexical level. And that means they deal with terms not words.

Figure 3.1 shows the categorization of the indexing algorithms.

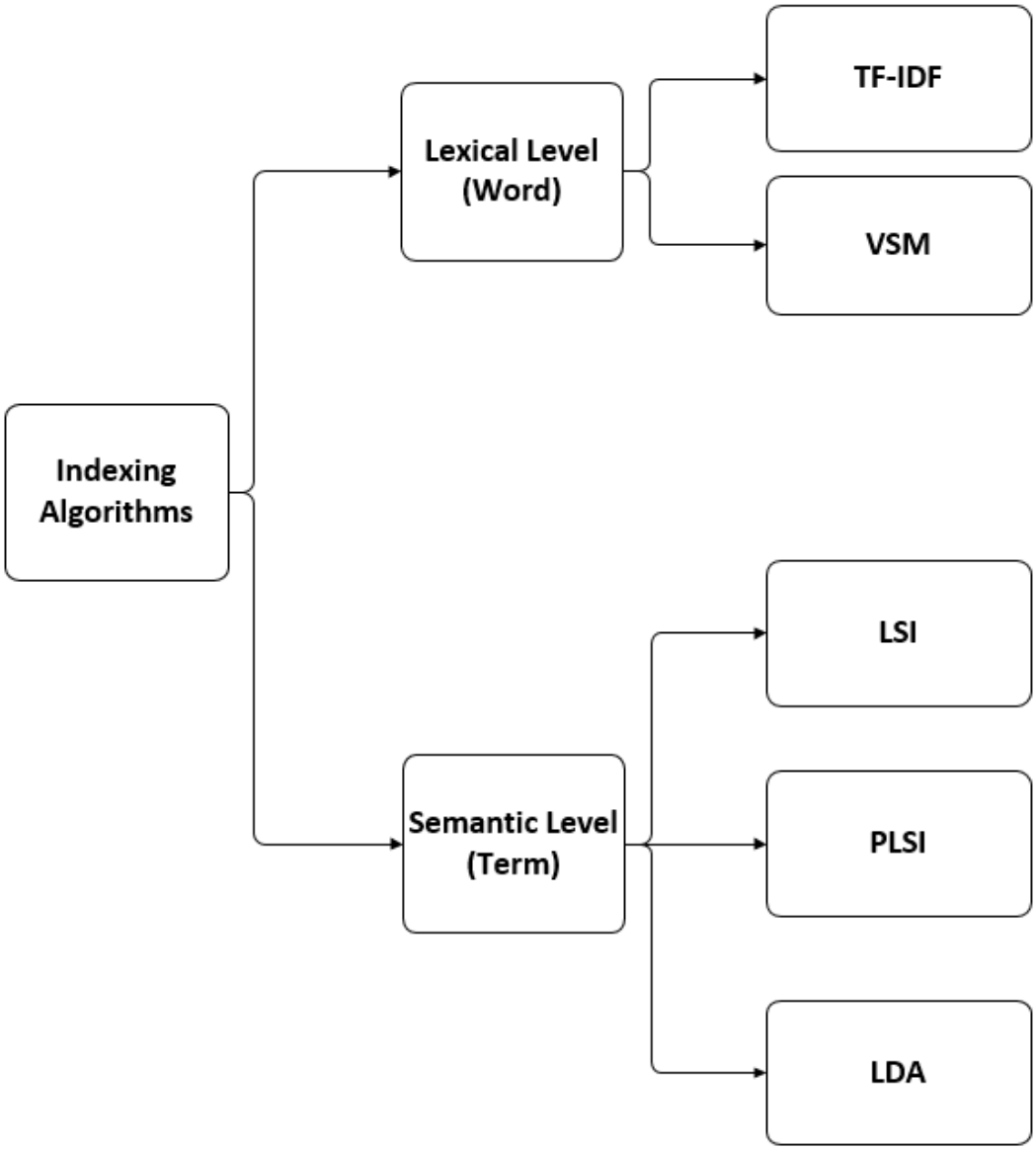


Figure 3.1 Categorization of Indexing Algorithms

In the next section, an overview of each algorithm will be discussed in more details; such as the model, advantages and disadvantages of each one.

3.3 Indexing Algorithms

3.3.1 Term Frequency- Inverse Document Frequency (TF-IDF):

TF-IDF is a numeric statistic designed to reflect how important a word is for a document in a collection [40]. It is often used as a weighting factor when searching for information and intellectual analysis of the text. The TF-IDF

value increases in proportion to the number of times the word appears in the document [41], but is often offset by the frequency of the word in the case, which helps to adapt to the fact that some words appear more often in general. The mathematical equation for this technique is:

$$tf - idf_{t,d} = tf_{t,d} \times idf_t \tag{3.1}$$

Where $tf - idf_{t,d}$ is the score between query t and document d. $tf_{t,d}$ is the term frequency and idf_t is the Inverse Document Frequency.

TF-IDF is one of the most popular term-weighting schemes. Variations of the TF-IDF weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

3.3.2 Vector Space Model (VSM):

VSM is an algebraic model for representing text documents and other objects containing linguistic semantics for search [42]. VSM represents terms as identifier vectors (indices). This model is used for filtering information, searching, indexing and ranking the relevance. VSM is often used with TD-IDF for weighing.

In VSM, documents and queries are represented in the form of vectors. The way to determine which document is similar to another document or to a query is to calculate the cosine of the angle between the vectors representing.

The similarity equation is:

$$sim\ (d1, d2) = \cos\theta = \frac{\vec{V}\ (d1).\ \vec{V}(d2)}{\left|\vec{V}\ (d1)\right|\left|\vec{V}(d2)\right|} \tag{3.2}$$

The query score is the similarity between document d and query q can be computed using the following equation:

$$sim\ (d, q) = \cos\theta = \frac{\vec{V}\ (d).\vec{V}(q)}{\left|\vec{V}\ (d)\right|\left|\vec{V}(q)\right|} \tag{3.3}$$

where the norm of the vector is:

$$\|q\| = \sqrt{\sum_{i=1}^{n} {q_i}^2} \quad (3.4)$$

3.3.3 Latent Semantic Indexing (LSI):

LSI is another natural language processing method which is used to discover information about the meaning of words [43]. LSI analyzes the relationship between the set of documents and the terms contained in them and assumes that words similar in meaning will occur in similar fragments of the text. LSI then constructs a matrix of words (terms) per document and uses the singular value decomposition (SVD) to reduce and divide the large matrix into small orthogonal components. Finally, present the word vectors in the documents.

The process of LSI is as following:

Given documents d1,…,dm and vocabulary words w1,…,wn, we construct a document-term matrix X∈Rm×n where xij describes the occurrence of word wj in document di. (For example, xij can be the raw count, 0-1 count, or TF-IDF.)

The dot product of row vectors is the document similarity, while the dot product of column vectors is the word similarity.

To reduce the dimensionality of X, truncated SVD is applied:

$$X \approx U_t \sum_t V_t^T \quad (3.5)$$

Each column of $U_t \in R_{mxt}$ and $U_t \in R_{mxt}$ corresponds to a document topic.

Now we can find similarity between w_i and w_j by finding the dot product of rows *i* and *j* of V_t.

And find documents relevant to the search query $d*$ by applying the SVD mapping on $d*$ and taking dot products with the rows of U_t.

3.3.4 Probabilistic Latent Semantic Indexing (PLSI):

PLSI is a statistical method for analyzing co-occurring data [44]. Compared with the standard latent semantic analysis that results from linear algebra and reduces the appearance tables usually through the expansion of singular

values, PLSI is based on the decomposition of the mixture obtained from the hidden class model. Instead of matrices, PLSI uses probability methods to represent semantics. Instead of using matrices, PLSI uses the probabilistic method. Its model is:

$$P(w|d) = P(d) \sum_{c} P(c|d)P(w|c) \tag{3.6}$$

Where d is the document index, c is word's topic drawn from $P(c|d)$, and w is word drawn from $P(w|c)$. And both P(c|d) and P(w|c) are modeled as multinomial distributions.

PLSI has some problems, such as it does not provide probabilistic model at the level of documents. In addition, the number of the parameters in the model increases linearly with increasing of the size of the document collection. PLSI does not explain clearly the way of assigning probability to a document in environment outside the training data.

3.3.5 Latent Dirichlet Allocation (LDA):

LDA is an improvement to PLSI, summarizing it using Dirichlet Prior, because the variable reflects the normal distribution of words in documents [45].

LDA assumes that each document contains different topics, and words in the document are generated from these topics. All documents contain a specific set of topics, but the proportion of each topic in each document is different. Figure 3.2 shows the graphical representation of LDA model.

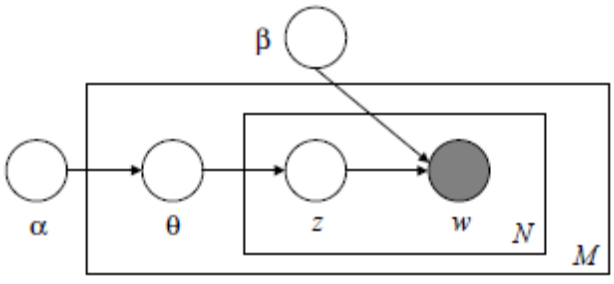


Figure 3.2 LDA Model

The generative process of the LDA model can be described as follows [46].

assuming document *w* in a corpus *D*:

1- Choose N ~ Poisson (ξ).

2- Choose θ ~ Dir(a).

3- For each of the *N* words w_n:

(a) Choose a topic z_n ~ multinomial distribution (θ)
(b) Choose a word w_n from $p(w_n|z_n,\beta)$, a multinomial probability conditioned on the topic z_n.

Many simplifying assumptions are made in this basic model, such as removing some subsequent sections.
First, the dimensionality *k* of the Dirichlet distribution which means that the dimensionality of the topic variable *z* is assumed to be known and fixed. Second, the word probabilities are parameterized by k $k \times V$ matrix β where $\beta_{ij} = p(w^j = 1 \,|z^i = 1)$ which for now we treat as a fixed quantity that is to be estimated.
Finally, the Poisson assumption is not critical to anything that follows and more realistic document length distributions can be used as needed.
Furthermore, note that *N* is independent of all the other data generating variables (θ and *z*). It is thus an ancillary variable and we will generally ignore its randomness in the subsequent development. A *k*-dimensional Dirichlet random variable θ can take values in the (*k*−1)-simplex (a *k*-vector θ lies in the (*k*−1)-simplex if $\theta_i \geq 0, \sum_{i=1}^{k} \theta_i = 1$), and has the following probability density on this simplex:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \dots \theta_k^{\alpha_k - 1} \tag{3.7}$$

Where the parameter α is a *k*-vector with components $\alpha_i > 0$, and where $\Gamma(x)$ is the Gamma function.
The Dirichlet is a convenient distribution on the simplex—it is in the exponential family, has finite dimensional sufficient statistics, and is conjugate to the multinomial distribution.

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w is given by:

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \tag{3.8}$$

Where $p(z_n|\theta)$ is simply θ_i for the unique i such that$z_n^i = 1$. Integrating over θ and summing over z, we obtain the marginal distribution of a document:

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta \tag{3.9}$$

Finally, the probability (or the log-likelihood) of generating corpus is:

$$p(D|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta) p(w_{dn}|z_{dn}, \beta) \right) d\theta_d \tag{3.10}$$

3.4 Comparison of Indexing Algorithms

Based on [46] and [47] research studies and simulation results Figure 3.3 shows the comparison between the indexing algorithms discussed in this section.

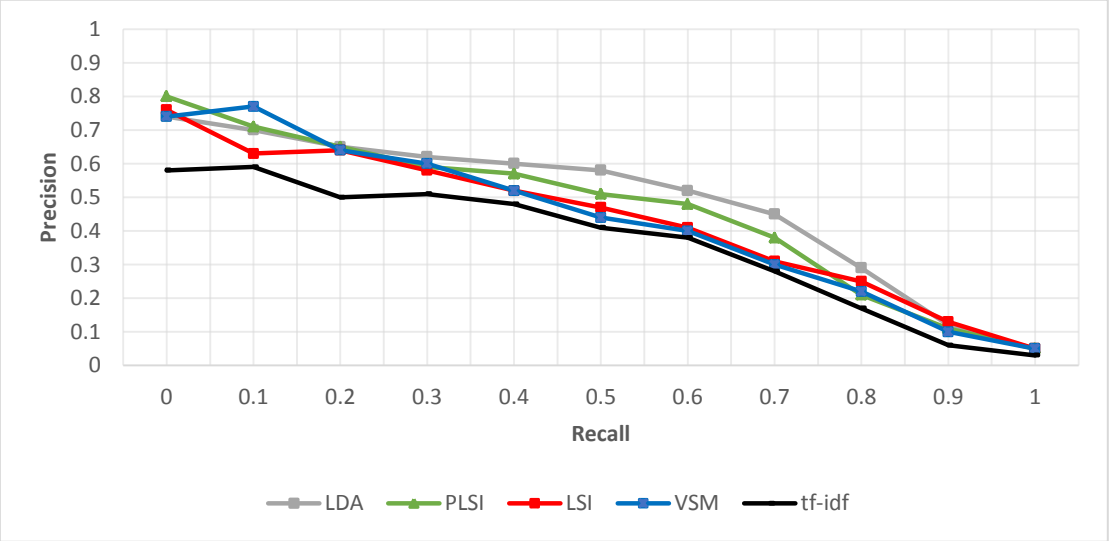


Figure 3.3 Comparison between Indexing Algorithms [46] [47]

As shown in Figure 3.3 LDA shows the best result for precision vs. recall ratio, although VSM shows in a small interval at the beginning of indexing better results than LDA.

Also LDA has the best performance between the algorithms mentioned in this chapter and that means LDA is the best option for document indexing.

3.5 Summary

As previous researches compared between semantic indexing algorithms LDA was the best according to the quality of output, which can be measured by precision and recall.

In this thesis, the model has been proposed – will be discussed in the next chapter (Chapter 4) -for semantic social network based on multi-agent concept with main two agent roles: indexing, which is done using LDA and ranking, that is done using Tag Rank.

In this thesis, we will use LDA algorithm and we will find the best output of LDA indexing process based on modifying the parameters affecting the result.

In addition, we will apply filter to cut the low results of indexing to refine the output and maintain higher precision and recall which means increasing the integrity and relevancy of the data.

Chapter Four: The Proposed Model for Multi-Agent Semantic Social Networks based on Tag Rank (MSSNT)

Chapter Four:

The Proposed Model for Multi-Agent Semantic Social:

4.1 Introduction

In this chapter, we introduce our proposed model of the semantic social network (MSSNT) which is multi-agent model based on network perspective where agents carry out specific centralized roles in SSN.

The input in this model is the document collection where it contains the word per document count. Then the final output will be the ranking of the tags, which are the Tag Rank results of the topics index.

The proposed model is based on two main phases: the indexing phase which is carried out by the indexing agent, and the ranking phase which is carried out by the ranking agent.

In the indexing phase, the input is the document collection where it contains word and document count. In this phase, the initialization is done then document parsed to get the initial index to be processed by (LDA) algorithm. The output of this phase is the semantic index, which contains word-per-topic distribution and topic-per-document distribution.

In our proposed model, we have focused on the topic-per-document to be used as tags. Therefore, in the next phase, which is the ranking phase the input will be the topic-per-document distribution that came as index matrix. In ranking phase, the input will be processed by Tag Rank algorithm. The final output will the Tag ranking matrix that will be sent to build the social links in the semantic social network.

4.2 System Architecture

Our (MSSNT) proposed model consists of the following:

- **Document Collection**: which are sets of raw data from social networks to be processed.
- **Indexing Agent**: In this part, three main processes are carried out; initializing documents, parsing document and indexing using semantic indexing algorithm.

-**Index**: It is the output of the document after indexer agent job completes. It contains the topic probabilities per document. And the word probability per topic. Topics are taken to be Tags for the first time. To be processed in ranking process.

- **P (Topic | document) (θ)**: This is the input of ranking process, which is extracted from the index output. In our proposed model that we propose that document topics will be the main tags to be ranked.

-**Ranking Agent:** In this part, we get the probabilities of topics per document. Then process them to be ranked as tags. Using certain ranking algorithm, which is based on self-learning feature which learns from training data which is updated after each ranking and indexing processes. Then gets the ranking output to make the links.

-**Social Graph**: The output of ranking agent will be used to build links between social nodes.

Figure 4.1 shows the proposed model architecture.

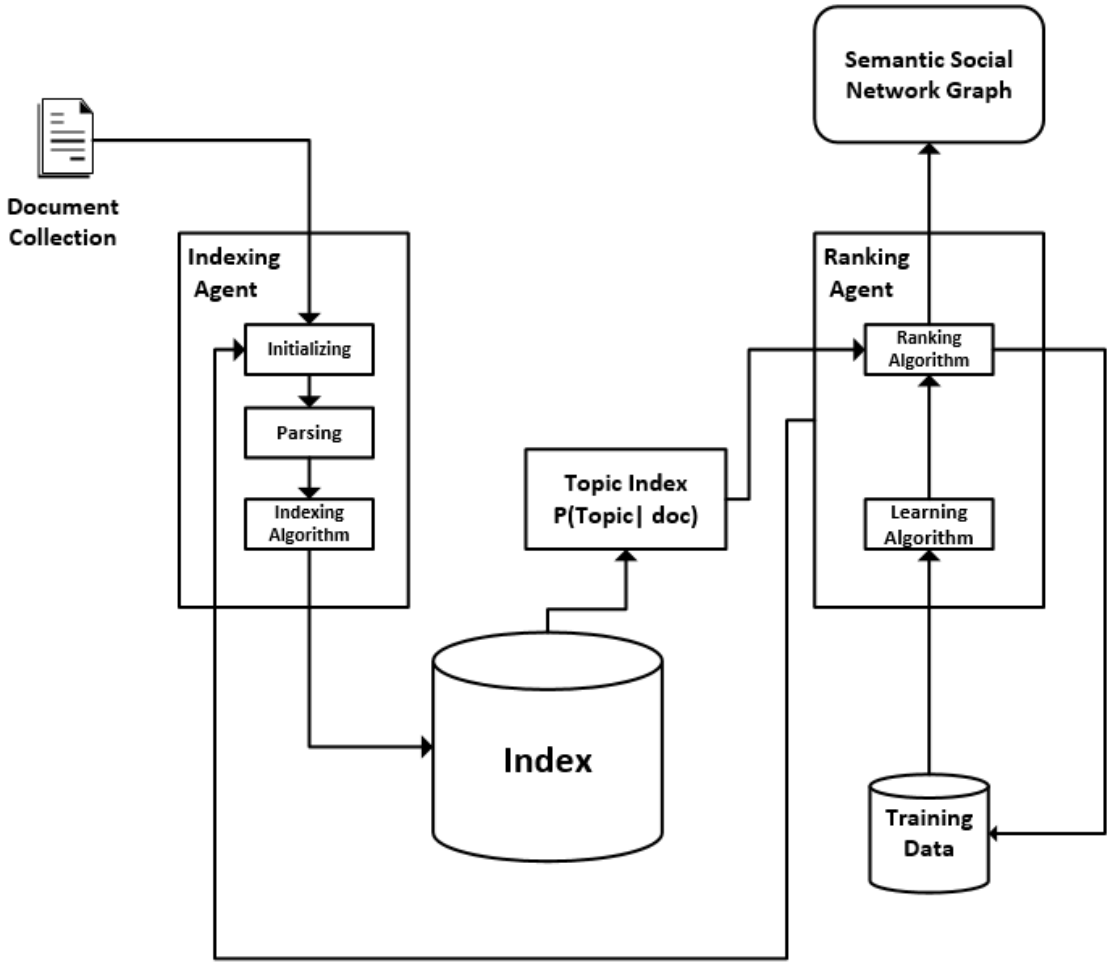


Figure 4.1 The Proposed System Architecture

As shown in Figure 4.1, the self-learning feature is implemented for both indexing and ranking agent, which means more intelligence in the proposed model. To explain the workflow of the proposed architecture, algorithms should be introduced to determine the roles of each agent in the architecture discussed earlier. The next section discusses these algorithms.

4.3 Algorithms

In this section, we will demonstrate the algorithms for our proposed model. Our proposed model (MSSNT) has two main phases: indexing phase, and ranking phase. The indexing phase has seven sequential steps to build the

topic per document index document based on the document collection to be processed. The steps of this phase are as follows:

1. **Input**: The system get the document collection as input which in this thesis will be obtained from dataset to do the simulation, also gets previous index output ($\theta^{\boldsymbol{I}}$)and previous rank output($\theta^{\boldsymbol{R}}$) as inputs.
2. **Parse**: The document collection is parsed to give every words specific identification number. In addition, words will be counted for each document. Moreover, documents will be given numbers. to do the indexing process.
3. **Index**: The process of indexing using (LDA) algorithm is done based on identification numbers and word count.
4. **Choose Optimal Parameters:** This sub-process will be carried out to determine the optimal parameters for (LDA) algorithm. For this algorithm, the parameters are the number of topics (k), and the indexing priors (a and β).
5. **Initial Index Output:** The result of LDA which is the topic per document index P(Topic| Doc) (θ) is optimized with the parameters, and has to be filtered in order to enhance the LDA output.
6. **Filter:** In this step, the index is filtered according to specific threshold (τ) to gain better precision and recall.
7. **Final Output:** Filtered P(Topic| Doc) (θ) is passed to ranking agent.

Figure 4.2 shows this flowchart explaining the steps of the indexing phase which is done by the indexing agent.

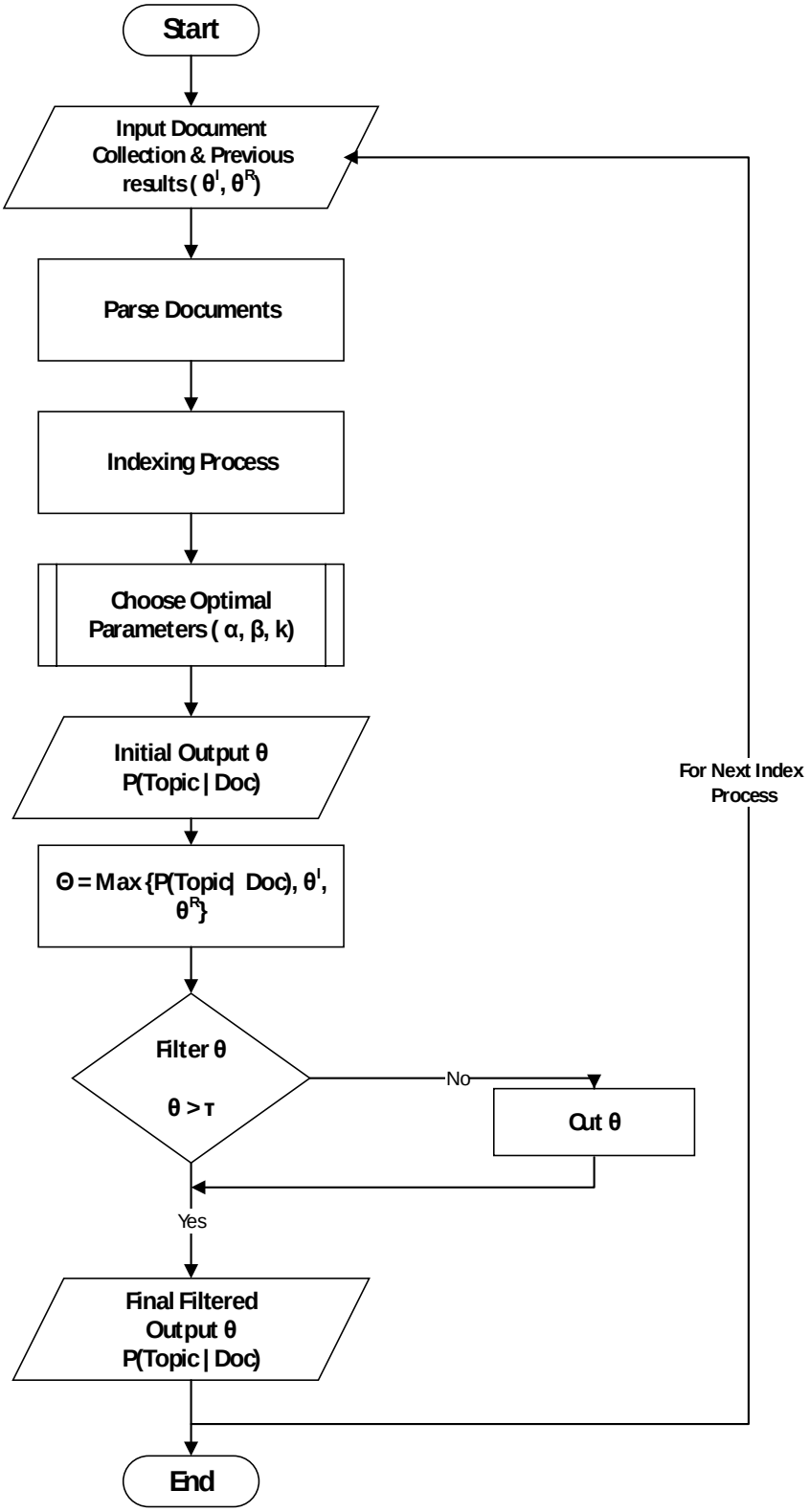


Figure 4.2 The Flowchart of the Indexing Phase

As shown in Figure 4.2, the start is with the input of document collection, which is parsed then indexed with choosing the optimal parameters (a, β *and* K) which increases the precision and recall of the output.

Then the output will be probability of topic per document that will be filtered by specific threshold (τ) that will be experimentally chosen in the next chapter (Chapter 5). The final output will be the filtered (θ)which is the output of our enhanced LDA algorithm. Which is called E-LDA.

Algorithm 4.1 shows the pseudocode of the role of the indexing agent.

Algorithm 4.1. Indexing Phase Algorithm

Input: Document Collection
Start
//Indexing Agent{
Rule 1: Get {Documents, Previous Index ($\boldsymbol{\theta^I}$), Previous Rank ($\boldsymbol{\theta^R}$)}
Rule 2: Parse Document Content
for *i=1* **to** *n* **do** //n= number of document records
{**Rule 3: Start** LDA Indexing Algorithm}
end for
Rule 4: Filter
{

for *i=1* **to** *n* **do** //n= number of document records
Select $\boldsymbol{\theta_{t_i} = Max\ (\theta_{t_i}, \theta^I, \theta^R)}$
Select $\boldsymbol{\theta_{t_i} where\ \theta_{t_i} > \tau}$ **// $\boldsymbol{\tau}$ is threshold**
end for

}
Output Index $\boldsymbol{(\theta_{t_1}, \theta_{t_2}, \dots, \theta_{t_n})}$
end }
//end of indexing agent job

As the indexing process is not quick process and it takes long time for big data like data for a semantic social network. Therefore, any documents created will not be indexed neither ranked until the next system scheduled indexing process. To overcome this issue, any new document is parsed and its words will be counted. Then for the most frequent words in this document, a comparison is made based on the word count to the most frequent words. And then the most frequent word is proposed as topic and it takes the weight as

average to its word count in the similar documents. Note that prepositions (such as: in, on, at, etc.) and articles (such as: the, a, an, …etc.) will not be considered as topics because indexing process deals with passed word without the prepositions or the articles. In the next section, we will discuss this patching solution equation in the discussion of the mathematical model.

The next phase to be discussed is the ranking phase. In this phase,

The steps of the ranking phase is as following:

1. **Input**: The input for this phase is the filtered P(Topic | Doc) (θ)
2. **Check Filter**: It ensures that all (θ) values are within the filtered values.
3. **Determine perspective of ranking**: in this step two tracks for processing is done. The tracks are:
    a. **Per Tag**: This track is to rank the documents according to specific Tag (topic).
    b. **Per Document**: This track is to process tags (topics) according to specific document.
4. **Compare and Maximize**: In this step comparison made between values of (θ) to determine the better Tag Rank score.
    a. **Per Tag**: For specific topic, all documents are compared to select the document with the maximum value of (θ) for this topic.
    b. **Per Document**: For specific document, all tags are compared to select the tag with the maximum value of (θ) for this document.
5. **Sort**:
    a. **Per Tag**: After comparison and maximization, a descending sorting to the documents with the same topic is done based on value of (θ).
    b. **Per Document**: Here also a descending sorting to the topics in a document is done based on value of (θ).
6. **Output**:
    a. **Per Tag**: Tag Rank output as document ranked order for specific tag.
    b. **Per Document**: Tag Rank output as tag ranked order for specific document.

Figure 4.3 shows the flowchart explaining the ranking phase.

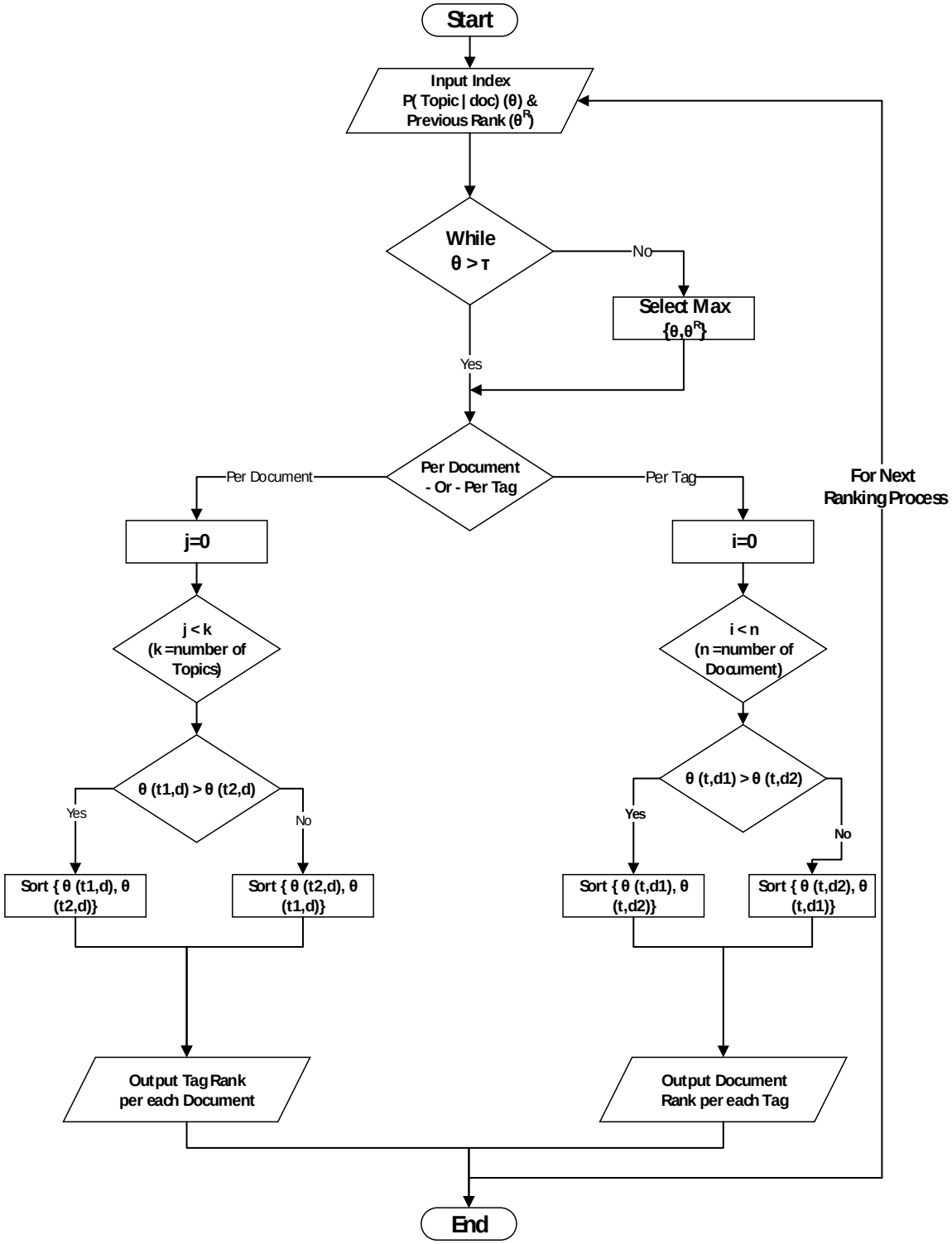


Figure 4.3 The Flowchart of the Ranking Phase

In Figure 4.3, the start is with the output of E-LDA algorithm with checking that (θ) is higher than the threshold (τ). Then the Tag Rank algorithm starts to rank (θ) as initial tag rank. The ranking algorithm is simply to maximize the

rank. Each document will get the higher topic ranking to be the first tag. In addition, documents will be descending ranked for each tag i.e for each topic.

Algorithm 4.2 shows the pseudocode of this phase.

Algorithm 4.2. Ranking Phase Algorithm

Input: Index ($\theta_{t_1}, \theta_{t_2}, \dots, \theta_{t_n}$), Previous Rank ($\theta^R$)
//Ranking Agent{
Start

for *i*=1 to *n* **do** //n= number of tags
//repeat until all tags which have larger ranks than threshold $\boldsymbol{\tau}$

Repeat{
//select document 1 and document 2 to be compared and maximized
θ_i =Max $\{ \theta_i, \theta_i^R \}$
θ_{i+1} =Max $\{ \theta_{i+1}, \theta_{i+1}^R \}$
Select Max(θ_i, θ_{i+1})

Condition: While (Max(θ_i, θ_{i+1}) $\geq \tau$) { // τ is threshold

Select Max(θ_i, θ_{i+1})

Sort (θ_i, θ_{i+1})
}
i=*i*+1
} // until (all tags which are larger than τ are processed).
for *j*=1 to *k* **do** //k= number of documents
//repeat until all documents which have larger ranks than threshold $\boldsymbol{\tau}$
Repeat{
//select tag 1 and tag 2 which are columns and rows of Max($\theta_{t_j}, \theta_{t_{j+1}}$)
Select Max($\theta_{t_j}, \theta_{t_{j+1}}$)
Condition: While (Max($\theta_{t_j}, \theta_{t_{j+1}}$) $\geq \tau$) { // τ is threshold

Select Max($\theta_{w_j}, \theta_{w_{j+1}}$)

Sort ($\theta_{w_j}, \theta_{w_{j+1}}$)

}

j=*j*+1

} // until (all tags which are larger than τ are processed).

Build Links between Tags

Output Tag Rank records

end } //end of Ranking Agent job.

As dealing with multi-agents, The AUML is used to illustrate the interactions of an agent-based system architecture. Figure 4.4 shows the sequence diagram model to show the interactions of the proposed model.

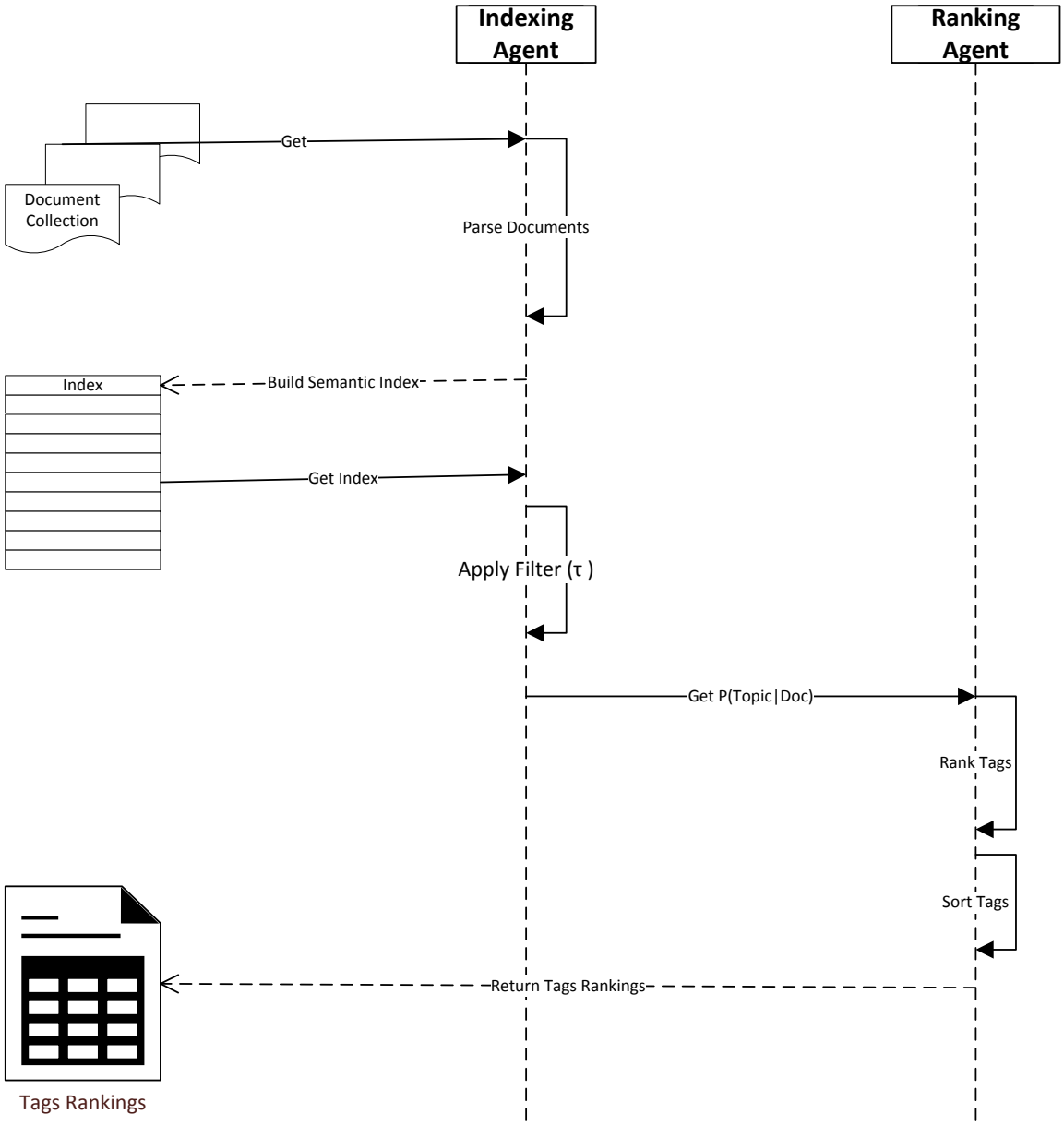


Figure 4.4 The System AUML Sequence Diagram

As shown in AUML diagrams, agents will be used to carry out the main two roles in our proposed model, which are indexing and ranking.

4.4 Mathematical Model

In this section, we will explain the mathematical model of our proposed model.

Based on derivation of LDA in section (3.3.5) we continue in our model.

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of *N* topics *z*, and a set of *N* words *w* is given by:

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta) \tag{4.1}$$

Where $p(z_n|\theta)$ is simply θ_i for the unique i such that $z_n^i = 1$. Integrating over θ and summing over *z*, we obtain the marginal distribution of a document:

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta)p(w_n|z_n, \beta) \right) d\theta \tag{4.2}$$

And focusing on topic per document distribution (P(topic| doc)) (θ) we can get our filter equation:

$$p(\theta, z, w|\alpha, \beta) = \begin{cases} p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta) & \theta \geq \tau \\ 0 & \theta < \tau \end{cases} \tag{4.3}$$

As we discussed in the previous section, between the two indexing processes may be new documents arises in (SSN) and so we have to initialize them to be ranked and linked temporarily before the next scheduled indexing process.

For that, we suppose that we have document (*d)* which has a word (*w*) with the highest word count. Also (*d*) has similar word count with (*d1*,*d2*,..*dn*) in the terms (*t1*, *t2*, ..*tn*) where (*n*) is thenumber of similar documents of (*d*)

The initial patch weight of (*w*) to be considered as topic is (θ_w). So, the equation of calculating (θ_w) is:

$$\theta_w = \frac{\theta_{t1,d1} + \theta_{t2,d2} + \cdots + \theta_{tn,dn}}{n} \quad (4.4)$$

And for (n) documents the general equation is:

$$\theta_w = \frac{\sum_{i=1}^{n} \frac{\theta_{ti,di}}{i}}{n} \quad (4.5)$$

Then from this equation, we will get (θ) to be as the initial tag weight that will be ranked:

For (k) which is the number of topics:

Maximize

$$\sum_{i=1}^{k} (\theta_i) \geq \tau \quad (4.6)$$

4.5 Summary

In this chapter, we have presented our model. Starting with an introduction, which gives a clear idea about the system. Then in second section, is the architecture of our model. In the third section, algorithms and flowcharts explaining processing flow is presented, also we use AUML to represent the Multiagent roles and interactions in our proposed model. After that the mathematical equations that is used in out model is provided.

Our model was represented by different methods to understand and simplify the overview of this proposed model. And also to prepare the proper implementation to be done in simulation chapter which is the next chapter.

Chapter Five: Simulation and Results

Introduction

Metrics for Evaluating Simulation

5.2.1 Metrics for Evaluating Indexing Algorithms

5.2.2 Metrics for Evaluating Ranking Algorithms

Data Set

Indexing Agent (LDA enhancements)

Ranking Agent- Tag Rank based on P(Topic | Doc) (θ)

Summary

Chapter Five:

Simulation and Results:

5.1 Introduction

In this chapter, we will present our simulation experiment for our proposed model. In this chapter, the concept of combining index resulting from LDA with threshold applied to be as the Tag input for the ranking algorithm has to be proven by results and providing a good comparison between our model in both indexing and ranking phases.

5.2 Simulation Tool

MATLAB is a powerful language for technical computing used by students, engineers and scientists in universities, research institutes and industries around the world [48]. The name MATLAB means **MAT**rix **LAB**oratory, because its main data element is a matrix (array).

MATLAB can be used for mathematical calculations, modeling and modeling, analysis and processing of data, visualization and graphics, and development of algorithms.

MATLAB is widely used in universities and colleges for introductory and advanced courses in mathematics, science and especially engineering. In the industry, software is used in research, development and design. Standard.

The MATLAB program has tools (functions) that can be used to solve common problems. In addition, MATLAB has additional toolbars, which are collections of specialized programs designed to solve specific problems. Examples include toolbars for signal processing, symbolic calculations and control systems.

5.3 Simulation Environment and Dataset

In this Thesis, simulation was carried out using MATLAB R2016a simulation software under Microsoft Windows 10 operating system.

The hardware platform that carried out the software is Intel core i7-3520M processor with 8 Gigabyte random access memory.

The simulation on the indexing phase will be carried out based on previous simulation works done by The Natural Language Processing Group at Stanford University [49], also on natural language labs on Iowa State University [50], and the research toolbox from University of California, Irvine [51], we use their MATLAB functions to implement our enhanced LDA function.

In this thesis, the dataset is used was *psychreview* dataset. Which contains Psychology Review Abstracts and collocation Data. This dataset contains about 85000 records of words and documents. With the initial count of words for each document and the topic.

5.4 Metrics for Evaluating Simulation

Based on section1.7 in Chapter1 we will consider the following metrics in our simulation.

1. **Precision**: It is the relation between the number of relevant documents retrieved to the total number of irrelevant and relevant documents retrieved, expressed usually as a percentage. The equation for calculating precision is:

$$Precision = \frac{|relevant\ documents \cap retreived\ documents|}{|retreived\ document|} \tag{5.1}$$

$$= \frac{true\ positive}{true\ positive + false\ positive} \tag{5.2}$$

2. **Recall**: It is the relation between the number of relevant documents retrieved to the total number of relevant documents in the dataset, expressed also as a percentage. The equation for calculating recall is:

$$Recall = \frac{|relevant\ documents \cap retreived\ documents|}{|relevant\ document|} \tag{5.3}$$

$$= \frac{true\ positive}{true\ positive + false\ negative} \tag{5.4}$$

Table 5.1 shows the relation between the two equations of precision and recall [49]:

Table 5.1 Precision and Recall Contingency Table

	Relevant	**Non Relevant**
Retrieved	True Positive	False Positive
Not Retrieved	False Negative	False Negative

These two metrics are used for evaluating indexing algorithms. The next two are used for evaluating ranking algorithms.

3. **Mean Average Precision (MAP):** It is the precision-at-*k* score of a ranking *y*, averaged over all the positions *k* of relevant documents.

   The equations for calculating (MAP) are:

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{5.5}$$

   Where:

$$AveP = Average\ Precison = \frac{\sum_{k=1}^{n}(P(k) \times rel\ (k))}{number\ of\ relevant\ documents} \tag{5.6}$$

   And *Q* is the number of queries, and:

$$rel(k) = \begin{cases} 1, & when\ item\ at\ rank\ (k) is\ relevant \\ 0 \end{cases} \tag{5.7}$$

4. **Normalized Discounted Cumulative Gain (NDCG):** It is a normalization of the Discounted Cumulative Gain (DCG) where (DCG) is a weighted sum of the relevancy degree of the ranked items. The weight is a decreasing function of the rank of the object, and therefore called discount.

The equations for calculating (NDCG) are:

$$NDCG_p = \frac{DCG_p}{IDCG_p} \tag{5.8}$$

Where:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2\, i} \tag{5.9}$$

And $IDCG_p$ is the ideal DCG at position p

In this chapter, we will evaluate our simulation with these metrics, showing the contributions we presented and how our proposed model is a good enhancement.

In the next section, we will discuss the enhancements were made in the indexing algorithm to be used for the indexing agent in our proposed model.

5.5 Indexing Agent (LDA Enhancements)

In this section, we will demonstrate the enhancements and optimizations done on LDA indexing algorithm. And discuss the results that is obtained from this simulation.

Here we are discussing about two main contribution: first is the optimization of the parameters of LDA process which are: α,β, and *k*.

For the used dataset, we have the true label of every document. Therefore, a guess of *k* would be the number of classes. The search for all three parameters requires too much work, so we first fix α and β to find the best number of classes *k*, and then with the best *k* we fix one of the other two remaining parameters to find the best value of the other that produces the best precision and recall values.

In this simulation, we select the test values as follows and based on [50]:

$\alpha = \{\frac{0.01}{K}, \frac{0.1}{k}, \frac{0.3}{\mathrm{k}}, \frac{0.5}{\mathrm{k}}, \frac{0.7}{\mathrm{k}}, \frac{1}{k})$

And $\beta = \{0.1, 1, 2\}$

And k= {2, 3, 4, 5, 6}

First, we fix the values of α ($\alpha = \frac{0.1}{k}$) and β ($\beta = 0.1$) to find the best value of k. The results were as shown on Figure 5.1.

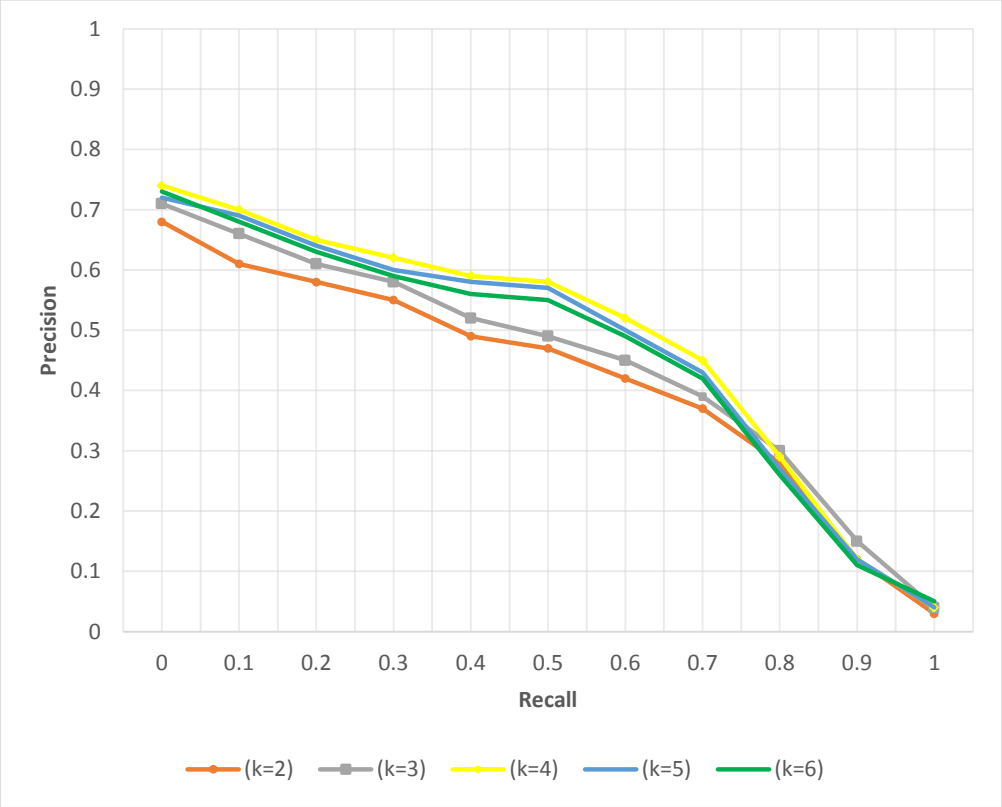


Figure 5.1 Precision vs. Recall for varying *(k)*

As shown in Figure 5.1. When k=4 (Yellow Line) we have the best precision and recall combination, which means when we classify collection into 4 topic classes we get the best results in indexing for the dataset used in this simulation.

With fixing k value (k=4) and β ($\beta = 0.1$) we have tested with varying α. Figure 5.2 shows the results:

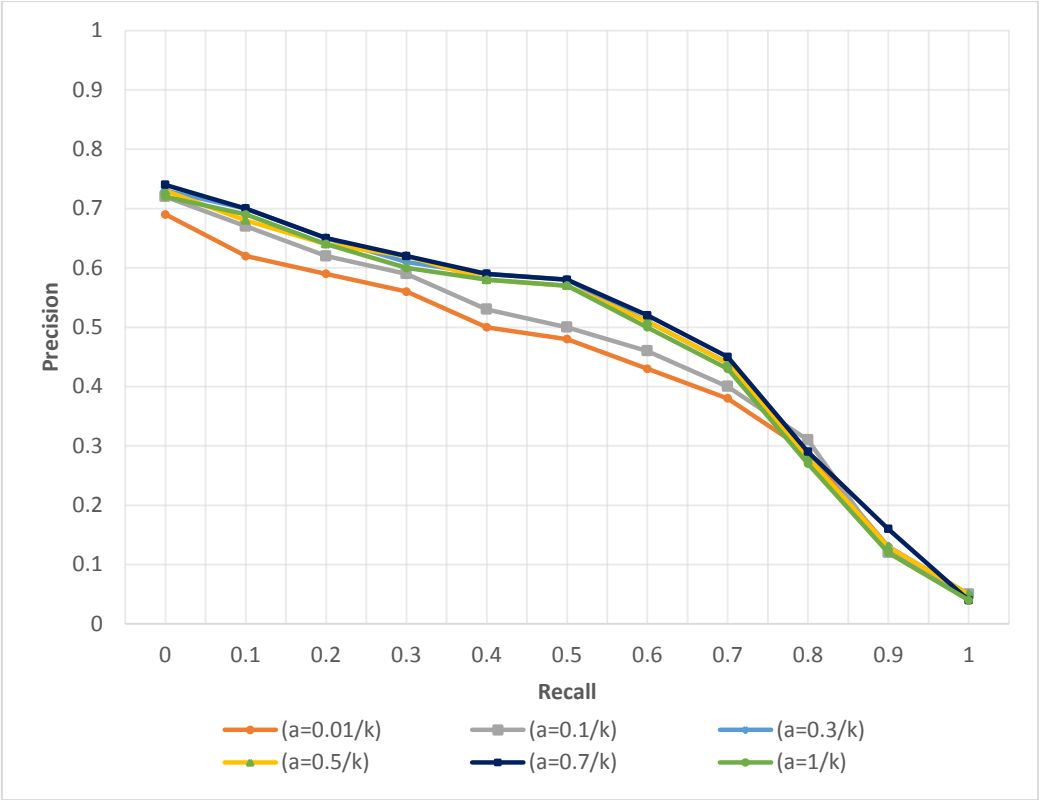


Figure 5.2 Precision vs. Recall for varying (α)

As shown in Figure 5.2 when $\alpha = \frac{0.7}{k} = \frac{0.7}{4} = 0.175$, we get the best result.

The final parameter to be determined is β, we have fixed k value (k=4) and α ($\alpha = 0.175$) to test varying β . Figure 5.3 shows the results:

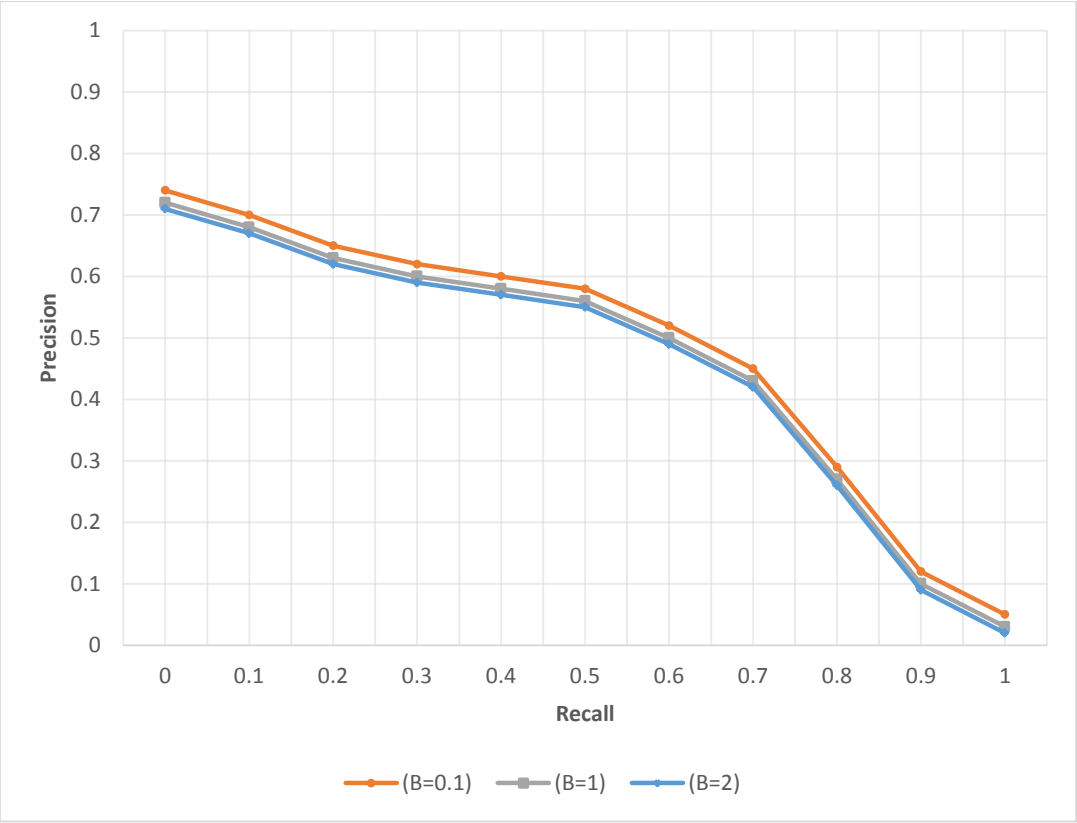


Figure 5.3 Precision vs. Recall for varying (β)

As shown in Figure 5.3, when $\beta = 0.1$, we get the best precision vs. recall results for the used dataset.

From previous scenarios, we see that the best result of LDA is when we choose the parameters for the used dataset as follows:

$k = 4,\ \alpha = \frac{0.7}{k}$, And $\beta = 0.1$.

The output of the indexing process will be in the form of documents and the related topics distribution probability. Table 5.2 shows an example of the output of the data between the documents number 1100 and number 1120.

Table 5.2 The Topic-per-Document Index for Documents (1100-1120)

	P(Topic 1| doc)	P(Topic 2| doc)	P(Topic 3| doc)	P(Topic 4| doc)
D1100	0.579888268	0.039851024	0.002607076	0.058472998
D1101	0.669135802	0.247930283	0.001016703	0.001016703
D1102	0.495302013	0.025503356	0.003131991	0.428187919
D1103	0.068808568	0.082195448	0.001874163	0.015261044
D1104	0.412290503	0.039851024	0.300558659	0.244692737
D1105	0.421105528	0.019095477	0.136348409	0.337353434

D1106	0.485115304	0.170649895	0.317400419	0.023899371
D1107	0.491517324	0.00167264	0.109199522	0.097252091
D1108	0.121765296	0.402607823	0.031494483	0.392577733
D1109	0.217130621	0.002997859	0.259957173	0.024411135
D1110	0.375035868	0.116786227	0.303299857	0.045050215
D1111	0.378674352	0.119308357	0.234582133	0.263400576
D1112	0.003203661	0.003203661	0.186270023	0.735469108
D1113	0.245380875	0.002269044	0.731604538	0.018476499
D1114	0.004560261	0.03713355	0.721172638	0.004560261
D1115	0.019757366	0.383708839	0.210398614	0.383708839
D1116	0.541829085	0.152023988	0.182008996	0.002098951
D1117	0.917691343	0.039397742	0.001756587	0.02685069
D1118	0.270484581	0.182378855	0.006167401	0.534801762
D1119	0.682225657	0.002163833	0.017619784	0.20309119
D1120	0.565339578	0.003278689	0.003278689	0.26088993

Table 5.2 shows the topic distribution for a small sample of the document collection. Figure 5.4 shows the distribution of topics to the whole collection.

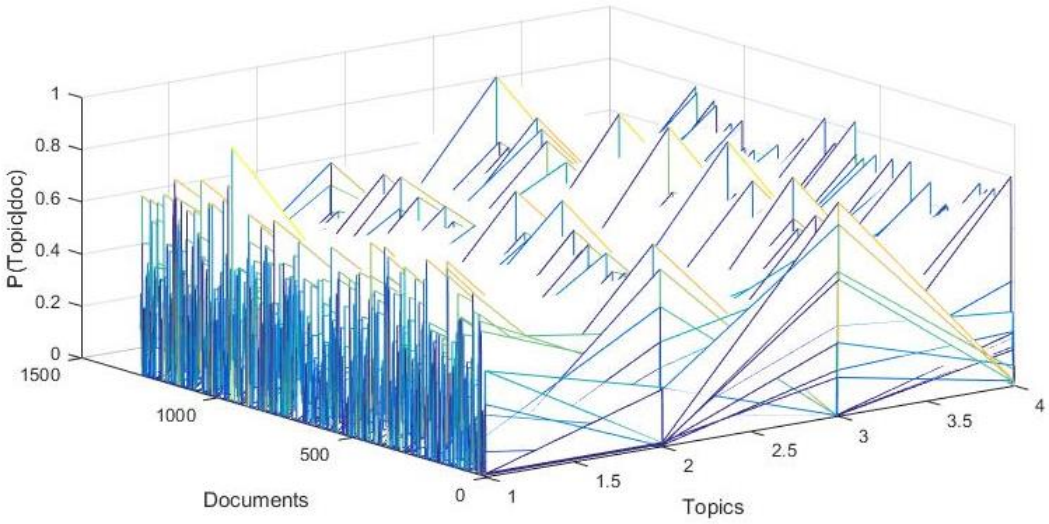


Figure 5.4 Topic Distribution in Document Collection

As shown in Figure 5.4 the topics are interrelated, thus this output need another enhancement, which is by using filter to the output of the indexing phase.

The second enhancement is to choose the best threshold (τ) to filter the output of the indexing process. Therefore, for the index output with the parameters that have been chosen before, we try to calculate the precision and recall of

the output with applying the filter. The result was as shown in Figures 5.5. And 5.6:

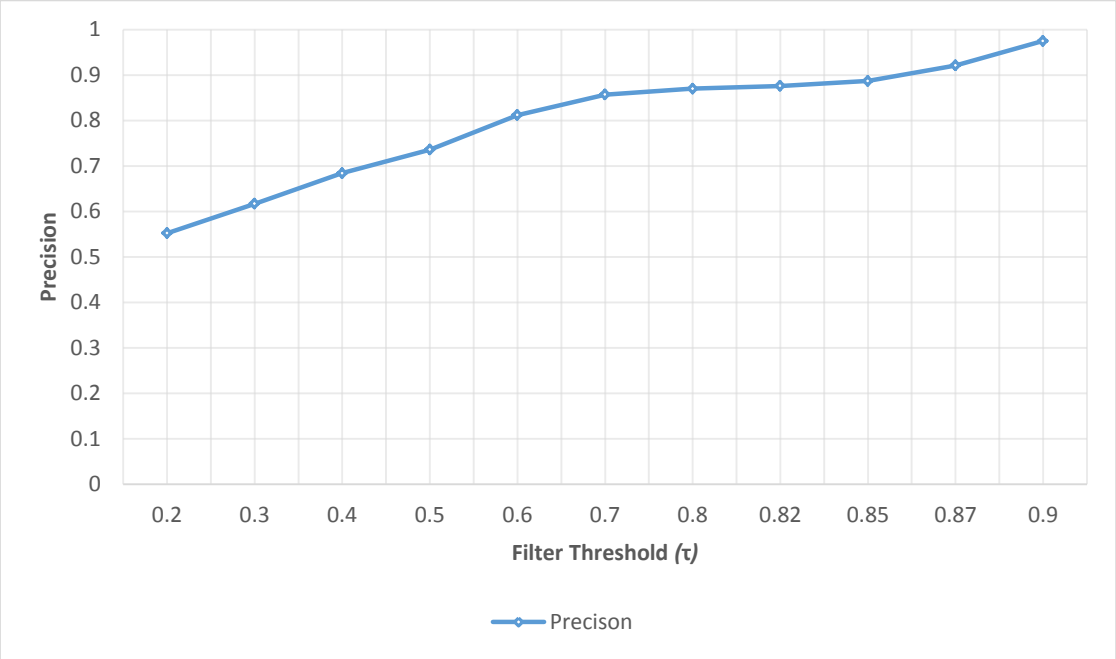


Figure 5.5 Precision according to (τ)

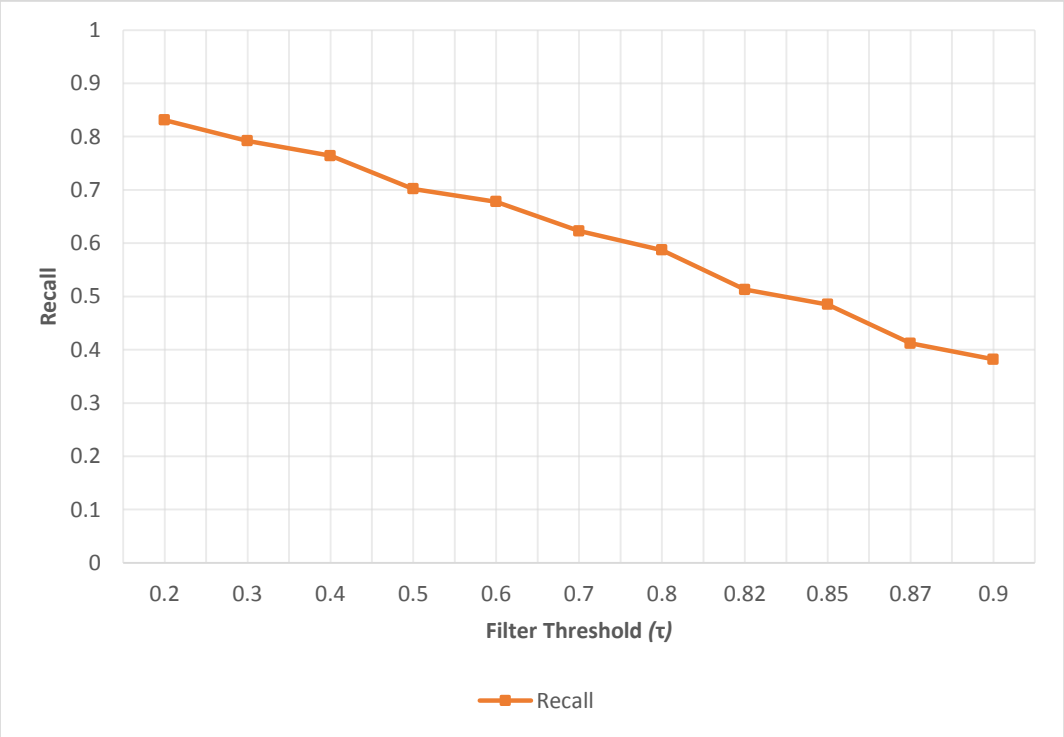


Figure 5.6 Recall according to (τ)

As shown in Figure 5.5. We notice that precision increases with increasing the threshold, while in Figure 5.6 we notice that recall decreases. The reason is that the relevant document varies with filter variation. Figure 5.7 shows the precision and recall curves according to variation of the filter threshold (τ).

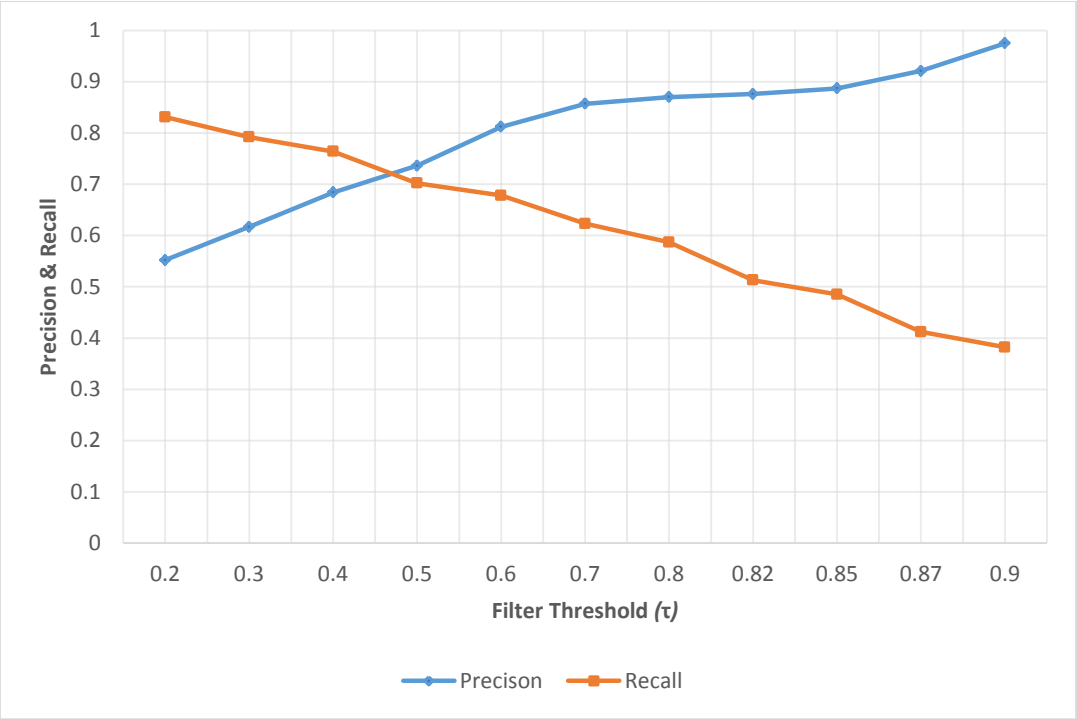


Figure 5.7 Precision and Recall according to (τ)

In Figure 5.7, it was noticed that the best combination of precision and recall is around τ=0.5. And so it is a good suggestion to choose this value as the threshold of the filter.

Comparing to Figure 5.4 the topic distribution will be after the filter as shown in Figure 5.8:

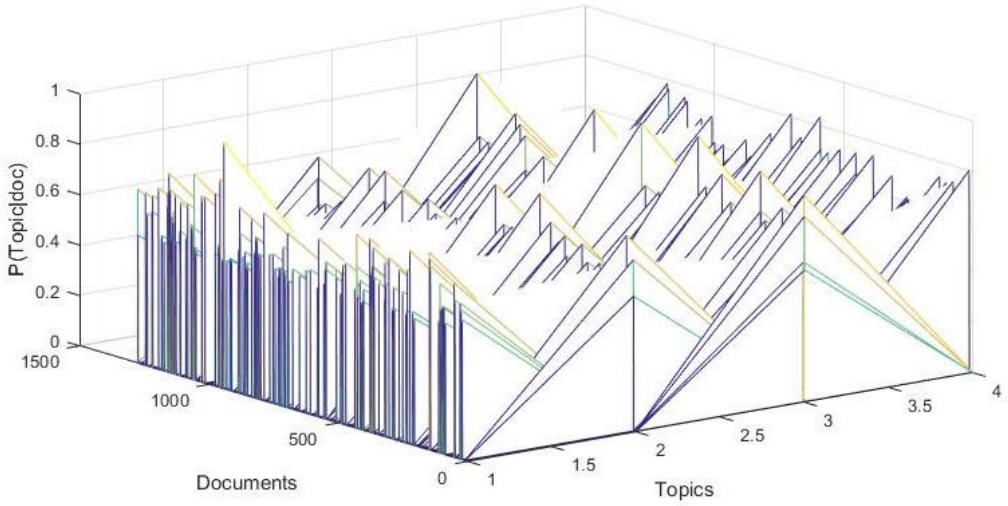


Figure 5.8 Topic Distribution in Document Collection after Filter

As shown in Figure 5.8, we have noticed that the topic distribution per documents is optimized and this means that the input for ranking phase is less and more relevant, which means that the result of the ranking phase will be more efficient. The result of these enhancements is represented as precision vs. recall curve, which is shown in Figure 5.9.

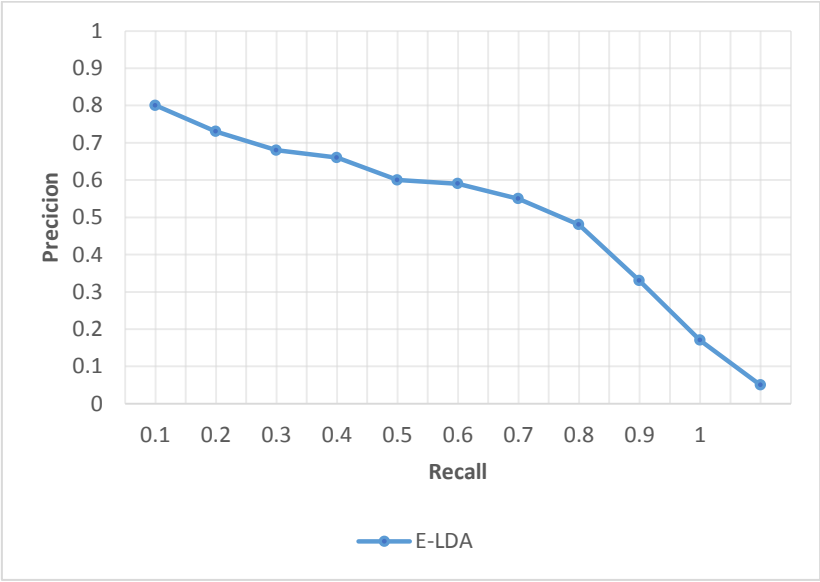


Figure 5.9 The Enhanced LDA Precision vs. Recall

As shown in Figure 5.9 we got a precision vs. recall curve in the enhanced LDA (E-LDA). In addition, that precision vs. recall ratio is better than LDA algorithm without the enhancements. Figure 5.10 shows the comparison between (E-LDA) and (LDA)

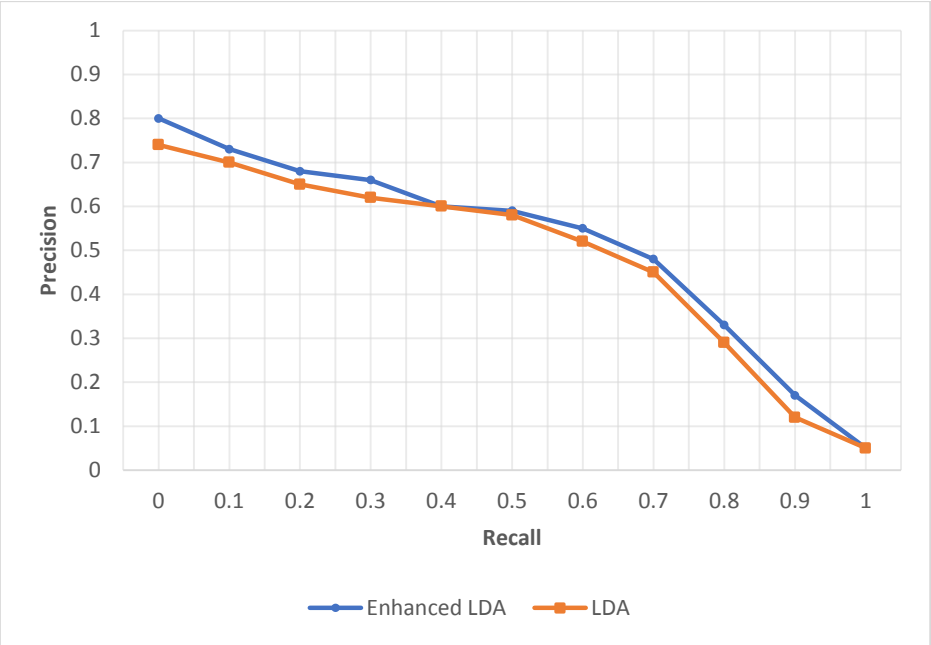


Figure 5.10 E-LDA vs LDA

As we discussed in Chapter 3, we showed that LDA is better that PLSI, LSI and the other semantic indexing algorithms.

With the results shown in Figure 5.9 we show that E-LDA is a good enhanced algorithm which is better than the other indexing algorithm which we have discussed in Chapter 3.

To ensure our results, we make simulation for the indexing algorithms based on researches discussed in Chapter3 [46] [47]. Table 5.3 shows the results of precision vs. recall of these algorithms on the data set used in our simulation.

Table 5.3 Precison vs. Recall for Indexing Algorithms

TFIDF		VSM		LSI		PLSI		LDA		E-LDA	
Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
0.58	0	0.74	0	0.76	0	0.8	0	0.74	0	0.8	0
0.59	0.09	0.77	0.13	0.63	0.11	0.71	0.12	0.7	0.122	0.73	0.125
0.5	0.19	0.64	0.22	0.64	0.22	0.65	0.24	0.65	0.25	0.68	0.26
0.51	0.29	0.6	0.31	0.58	0.31	0.59	0.32	0.62	0.32	0.66	0.33
0.48	0.38	0.52	0.405	0.52	0.41	0.57	0.42	0.6	0.43	0.6	0.44
0.41	0.49	0.44	0.5	0.47	0.5	0.51	0.52	0.58	0.53	0.59	0.55
0.38	0.57	0.4	0.59	0.41	0.6	0.48	0.61	0.52	0.62	0.55	0.64
0.28	0.66	0.3	0.68	0.31	0.7	0.38	0.71	0.45	0.72	0.48	0.73
0.17	0.75	0.22	0.78	0.25	0.79	0.21	0.8	0.29	0.81	0.33	0.82
0.06	0.84	0.1	0.87	0.13	0.88	0.11	0.89	0.12	0.9	0.17	0.91

Also Figure 5.11 shows comparison between E-LDA and the other indexing algorithms using precision vs. recall curves.

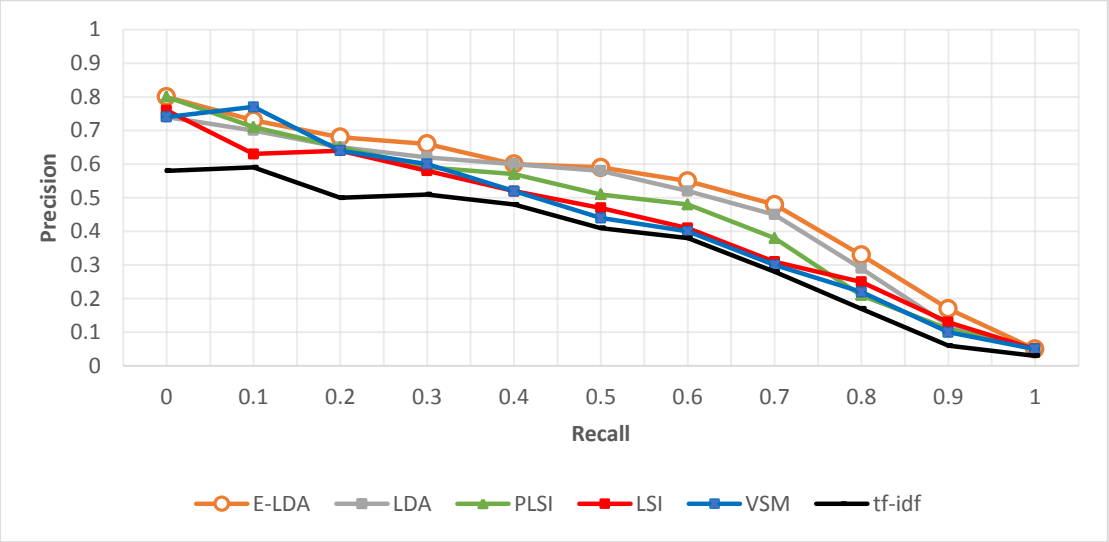


Figure 5.11 E-LDA vs. semantic indexing algorithms

As shown in Figure 5.9 E-LDA has better precision vs. recall combination which means better relevancy in index output.

5.6 Ranking Agent- Tag Rank based on P(Topic | Doc) (θ)

In this section, we get the output index from indexing agent and process it through Tag Rank process, which is simply comparing and choosing the maximum tag probability.

Figure 5.12 shows comparison of precision vs. recall curves before and after ranking process was carried out by the ranking agent.

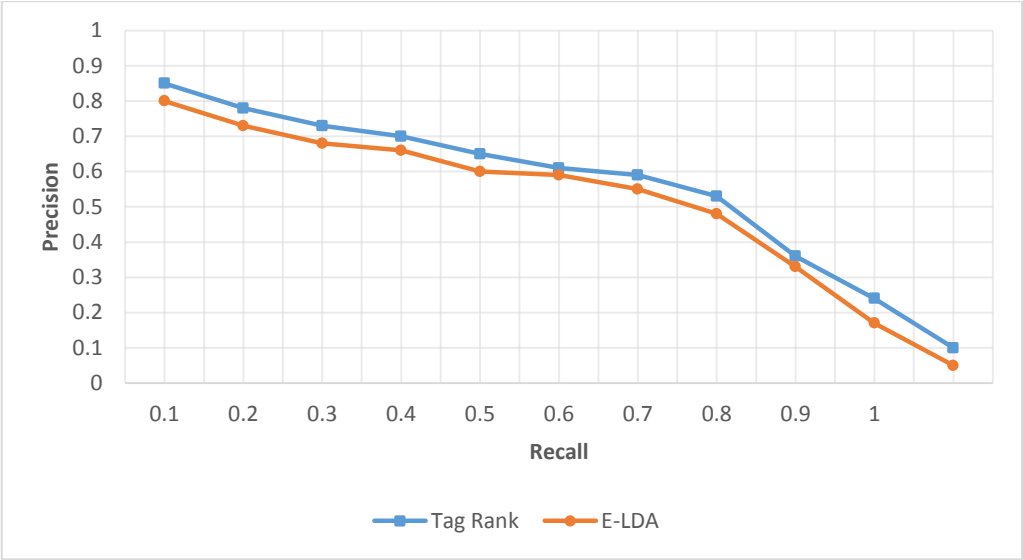


Figure 5.12 Precision vs. Recall according to ranking algorithm

As shown in Figure 5.12 the ranking process have increased the precision vs. recall combination and that should reflect true behavior of ranking agent as results will be enhanced after the ranking process was carried out as some tags probabilities will be maximized a little and so the precision and recall will be improved a little.

To compare the proposed ranking algorithm, we have done some comparison with some other algorithms discussed in Chapter 2 that can be applicable to the dataset we have used in our simulation.

Comparing Tag Rank with PR, WPR, HITS and TSPR (Time Rank) according to MAP and NDCG@(k=4) as (k=4) is the best parameter for the indexing algorithm LDA that we have concluded from the previous section. Table 5.4 shows the results of NDCG and MAP for these ranking algorithms.

Table 5.4 NDCG and MAP results for different Ranking algorithms

	NDCG @k=4	MAP
PR	0.52	0.3
WPR	0.54	0.31
HITS	0.53	0.29
TSPR	0.545	0.315
TagRank	0.56	0.32

And Figure 5.13 shows the comparison between ranking algorithms:

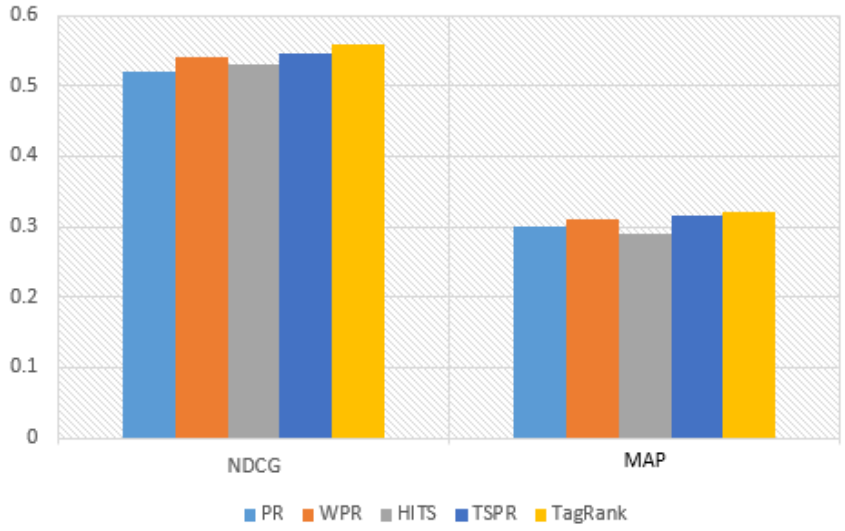


Figure 5.13 The Comparison between Ranking Algorithms

As shown in Table 5.4 and in Figure 5.13 Tag Rank shows the best MAP and NDCG values and so it could be said that Tag Rank is the best suitable ranking algorithm for this proposed model.

5.6 Summary

In this chapter, we provide our simulation experiments that were carried out in order to verify the enhancements of the proposed model of semantic social network. Our results and analysis are based on four metrics: Precision, Recall, Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG).

There were three scenarios for determining the best values of the three parameters of LDA algorithm (α,β, and k) on the dataset that have been tested in this simulation.

Then we have proposed the filter threshold (τ) of LDA index to improve relevancy. And by two scenarios of precision and recall we have determined the best value of threshold (τ).

As a summary of the results, "Enhanced LDA" shows a better performance as compared to other indexing algorithms.

Then we continue with ranking algorithm Tag Rank. Which shows an improvement of output of the indexing algorithm. Showing the improvement of precision and recall after Tag Rank phase.

Then we have compared Tag Rank with other ranking algorithms discussed in Chapter 2. We have shown that Tag Rank is the best algorithm in our simulation.

Chapter Six: Conclusion and Future Works

6.1 Thesis Conclusion

6.2 6.2 Future Works

Chapter Six:

Conclusion and Future Works:

6.1 Thesis Conclusion

Semantic Social Networks are new evolving topic that has many key marks that helps in improving new model for it. As data integrity and relevancy are the main challenge in this field, a lot have to be done to ensure and improve integrity and relevancy of data to be used.

This thesis aims mainly at providing new model of Semantic Social Network that is based on Multi-Agent Systems concept. This proposed model mainly consisted of two main agents: indexing agent that carries out enhanced Latent Dirichlet Allocation algorithm (E-LDA), and ranking agent that carries out Tag Rank algorithm.

Enhanced LDA (E-LDA) is distinguished from other preceding indexing algorithms and simulation results show an increase precision and recall of E-LDA. This algorithm's output is the topic index which can be the primary tag index to be processed by ranking agent.

Tag Rank is also distinguished from other ranking agents as it deals with tags that is more relevant to social networks and also more relevant to semantics. Simulation results show that Tag Rank produces better NDCG and MAP that other algorithms.

6.2 Future Works

In the future, we will add the term per topic index to be also entered as tags to be processed by ranking agent. This means that we will have larger data to be ranked. So, the processing conditions must be taken care of while implementing the system.

Also, Tag Rank algorithm will be developed and improved by representing another computations of ranking. Possible suggestion to use fuzzy logic ranking which will reflect more realistic agent roles. Because of the decision and the self-learning abilities that can be provided to the ranking agent.

Finally, and after all the future improvements, we can suggest building and implementing our model to a semantic social network. Either in an existing social network, or in new semantic social network programmed from the beginning based on our model.

References

[1] A. Obar, Jonathan & Wildman, Steve. (2015). Social media definition and the governance challenge: An introduction to the special issue. Telecommunications Policy. 39. 10.1016/j.telpol.2015.07.014.

[2] M. Boyd, Danah & Ellison, Nicole. (2007). Social Network Sites: Definition, History, and Scholarship. J. Computer-Mediated Communication. 13. 210-230. 10.1111/j.1083-6101.2007.00393.x.

[3] WonKim, Ok-RanJeong, Sang-WonLee. (2010) On social Web sites. Information Systems 35, 215– 236.

[4] Boyd, Dana; Crawford, Kate. (2011) Six Provocations for Big Data. Social Science Research Network: A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society. (September 21, 2011). doi:10.2139/ssrn.1926431.

[5] Downes, Stephen. (2005). Semantic networks and social networks. The Learning Organization. 12. 10.1108/09696470510700394.

[6] G. Zhang, C. Li, C. Xing and G. Zhang. (2012) A Semantic++ Social Search Engine Framework in the Cloud. Eighth International Conference on Semantics, Knowledge and Grids, Beijing, 2012, pp. 277-278. doi: 10.1109/SKG.2012.9.

[7] Michael Wooldridge. (2009). An Introduction to Multiagent Systems (2nd ed.). Wiley Publishing. ISBN:0470519460 9780470519462

[8] Kubera, Yoann & Mathieu, Philippe & Picault, Sébastien. (2010). Everything can be Agent!. Proc. of the 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS). 1547-1548. 10.1145/1838206.1838474.

[9] Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.

[10] Salamon, Tomas (2011). Design of Agent-Based Models. Repin: Bruckner Publishing. p. 22. ISBN 978-80-904661-1-1.

[11] Weyns, Danny & Omicini, Andrea & Odell, James. (2006). Environment as a First-Class Abstraction in Multiagent Systems. Autonomous Agents and Multi-Agent Systems. 14.

[12] Panait, Liviu & Luke, Sean. (2005). Cooperative Multi-Agent Learning: The State of the Art. Autonomous Agents and Multi-Agent Systems. 11. 387-434. 10.1007/s10458-005-2631-2.

[13] Stuart Russell and Peter Norvig. (2009). Artificial Intelligence: A Modern Approach (3rd ed.). Prentice Hall Press, Upper Saddle River, NJ, USA.

[14] M. Vadoodparast, Milad & Taghiyareh, Fattaneh. (2015). A multi-agent solution to maximizing product adoption in dynamic social networks. 71-78. 10.1109/AISP.2015.7123484.

[15] Wang, Yunlong & Djuric, Petar. (2015). Social Learning With Bayesian Agents and Random Decision Making. IEEE Transactions on Signal Processing. 63. 1-1. 10.1109/TSP.2015.2421486.

[16] Jiang, Yichuan & Zhou, Yifeng & Wang, Wanyuan. (2013). Task Allocation for Undependable Multiagent Systems in Social Networks. Parallel and Distributed Systems, IEEE Transactions on. 24. 1671-1681. 10.1109/TPDS.2012.249.

[17] Montañés, Elena & Quevedo, Jose & Díaz, I & Cortina, Raquel & Alonso, Pedro & Ranilla, Jose. (2011). TagRanker: Learning to recommend ranked tags. Logic Journal of the IGPL. 19. 395-404. 10.1093/jigpal/jzq036.

[18] Qian, Xueming & Lu, Dan & Liu, Xiaoxiao. (2016). Tag based Image Search by Social Re-Ranking. IEEE Transactions on Multimedia. 18. 1-1. 10.1109/TMM.2016.2568099.

[19] Qian, Xueming & Hua, Xian-Sheng & Tang, Yuan & Mei, Tao. (2014). Social Image Tagging with Diverse Semantics. IEEE transactions on cybernetics. 44. 2493-508. 10.1109/TCYB.2014.2309593.

[20] Franchi, Enrico. (2010). A Multi-Agent Implementation of Social Networks. Proceedings of the 11th WOA 2010 Workshop, Dagli Oggetti Agli Agenti, Rimini, Italy, September 5-7, 2010.

[21] Feng, Songhe & Feng, Zheyun & Jin, Rong. (2015). Learning to Rank Image Tags with Limited Training Examples. IEEE transactions on image processing: a publication of the IEEE Signal Processing Society. 24. 10.1109/TIP.2015.2395816.

[22] J. Zhang, Y. Yang, Q. Tian, L. Zhuo and X. Liu. (2017). Personalized Social Image Recommendation Method Based on User-Image-Tag Model," in IEEE Transactions on Multimedia, vol. 19, no. 11, pp. 2439-2449.

[23] William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus. (1992). Knowledge discovery in databases: an overview. AI Mag. 13, 3 (September 1992), 57-70.

[24] Järvelin, Kalervo & Kekäläinen, Jaana. (2017). IR evaluation methods for retrieving highly relevant documents. ACM SIGIR Forum. 51. 243-250. 10.1145/3130348.3130374.

[25] Rekha, Jain & Purohit, G.N.. (2010). Page Ranking Algorithms for Web Mining. International Journal of Computer Applications. 13. 10.5120/1775-2448.

[26] L. Bhamidipati, Narayan & Pal, Sankar. (2009). Comparing Scores Intended for Ranking. Knowledge and Data Engineering, IEEE Transactions on. 21. 21-34. 10.1109/TKDE.2008.111.

[27] Sharma, Duhan, and Kumar, G. (2010). A Novel Page Ranking Method based on Link- Visits of Web Pages. International Journal of Recent Trends in Engineering and Technology, Vol. 4, No. 1, pp 58-63.

[28] H Dunham, Margaret & Seshadri, Sridhar. (2006). Data Mining-Introductory and Advanced Topics. Pearson Education. ISBN 9788177587852

[29] Etzioni, Oren. (1997). The World-Wide Web: Quagmire or gold mine?. Communications of the ACM. 39. 10.1145/240455.240473.

[30] L. Page, S. Brin, R. Motwani, and T. Winograd. (1999). The PageRank

Citation Ranking: Bringing Order to the Web", Technical Report,

Stanford Digital Libraries SIDL-WP-1999-0120.

[31] Xing, Wenpu & Ghorbani, Ali. (2004). Weighted PageRank Algorithm. 305-314. 10.1109/DNSR.2004.1344743.

[32] M. Kleinberg, Jon. (1999). Authoritative Sources in a Hyperlinked Environment. Journal of The ACM - JACM. 46.

[33] Jiang, Hua & Ge, Yong-Xing & Zuo, Dan & Han, Bing. (2008). TimeRank: A method of improving ranking scores by visited time. 1654-1657. 10.1109/ICMLC.2008.4620671.

[34] Edge Rank website: http://edgerank.net/ [Accessed - May 1, 2018].

[35] Jie, Shen & Chen, Chen & Hui, Zhang & Rong-shuang, Sun & Yan, Zhu & Kun, He. (2008). TagRank: A New Rank Algorithm for Webpage Based on Social Web. 254-258. 10.1109/ICCSIT.2008.45..

[36] Hwang, DaeHoon. (2015). Comparison and Evaluation of Highly Related Tag-Pairs Extraction Methods. International Journal of Multimedia and Ubiquitous Engineering. 10. 353-362. 10.14257/ijmue.2015.10.9.36.

[37] Hwang, DaeHoon. (2016). Design of Tag Ranking Algorithm Based on Cluster. Advanced Science and Technology Letters Vol.133. Information Technology and Computer Science 2016, pp.194-200.

[38] Thirumala Sree Govada, and N Lakshmi Prasanna. (2014). Comparative study of various Page Ranking Algorithms in Web Content Mining (WCM). Int. J. of Adv. Res. **2** (7). 0] (ISSN 2320-5407)

[39] Sharma, Dilip & Sharma, Ashok. (2010). A Comparative Analysis of Web Page Ranking Algorithms. International Journal on Computer Science and Engineering. 2.

[40] D Manning, C & Raghavan, P & Schtextbackslashütze, H & Corporation, Ebooks. (2008). Introduction to Information Retrieval. ISBN 0521865719.

[41] Robertson, Stephen. (2004). Understanding Inverse Document Frequency: On Theoretical Arguments for IDF. Journal of Documentation - J DOC. 60. 503-520. 10.1108/00220410410560582.

[42] Turney, Peter & Pantel, Patrick. (2010). From Frequency to Meaning: Vector Space Models of Semantics. Journal of Artificial Intelligence Research. 37. 10.1613/jair.2934.

[43] Deerwester, Scott & T. Dumais, Susan & Furnas, George & Landauer, Thomas & Harshman, Richard. (1990). Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science. 41. 391-407. 10.1002.

[44] Hofmann, Thomas. (2017). Probabilistic Latent Semantic Indexing. ACM SIGIR Forum. 51. 211-218. 10.1145/3130348.3130370.

[45] M. Blei, David & Y. Ng, Andrew & Jordan, Michael. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research. 3. 993-1022. 10.1162/jmlr.2003.3.4-5.993.

[46] Wang, Yanshan & Lee, Jaesung & Choi, In-Chan. (2016). Indexing by Latent Dirichlet Allocation and Ensemble Model. Journal of the Association for Information Science and Technology. 67. 1736-1750. 10.1002/asi.23444.

[47] Choi, In-Chan & Lee, Jaesung. (2010) Document Indexing by Latent Dirichlet Allocation. In proceedings of the 2010 international conference on data mining, At Los Angeles, 2010. DMIN'10. 409-414.

[48] Gilat, Amos (2011). MATLAB: An Introduction with Applications 4th Edition. John Wiley & Sons. ISBN-13 978-0-470-76785-6

[49] The Natural Language Processing Group at Stanford University, https://nlp.stanford.edu/, accessed in May 1, 2018.

[50] Iowa State University, http://home.eng.iastate.edu, accessed in May 1, 2018.

[51] MATLAB Topic Modeling Research Toolbox in University of California, Irvine, http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm , Accessed in May 1, 2018.

Appendices

Appendix A: Acronyms and Abbreviations

SN	Social Network
UCC	User-Created Content
SSN	Semantic Social Network
MAS	Multi Agent System
LDA	Latent Dirichlet Allocation
MAP	Mean Average Precision
NDCG	Normalized Discounted Cumulative Gain
KDD	Knowledge Discovery in Database
IR	Information Retrieval
WWW	World Wide Web
WCM	Web Content Mining
WSM	Web Structure Mining
WUM	Web Usage Mining
PR	Page Rank
WPR	Weighted Page Rank
HITS	Hyper-link Induced Topic Search
TSPR	Topic Sensitive Page Rank
RSVP	ReSponse from the inVited Person
TWM	Tag-Pair Weight Matrix
TFM	Tag Frequency Matrix
TF-IDF	Term Frequency- Inverse Document Frequency
VSM	Vector Space Model
LSI	Latent Semantic Indexing
SVD	Singular Value Decomposition
PLSI	Probabilistic Latent Semantic Indexing
D	Document Collection
E-LDA	Enhanced Latent Dirichlet Allocation

Appendix B: Published Paper

Tag Ranking Multi-Agent Semantic Social Networks

Accepted and presented in the 2017 International Conference on Computational Science and Computational Intelligence (CSCI'17), held on December 14-16, 2017, in Las Vegas, USA.

http://americancse.org/events/csci2017

Tag Ranking Multi-Agent Semantic Social Networks

Rushdi Hamamreh
Computer Engineering Department,
Faculty of Engineering.
Al-Quds University.
rushdi@staff.alquds.edu

Sameh Awad
Computer Engineering Department,
Faculty of Engineering.
Al-Quds University.
sameh.awad@student.alquds.edu

Abstract—**Social Media has become one of the most popular platforms to allow users to communicate, and share their interests without being at the same geographical location. With the rapid growth of Social Media sites such as Facebook, LinkedIn, and Twitter, etc. There is vast amount of user-generated content. Thus, the improvement in the information quality has become a great Challenge to all social media sites, which allows users to get the desired Content or be linked to the best link relation using improved search / link technique. So introducing semantics to media networks will widen up the representation of the social media networks. Semantic Social Networks representation of social links will be extended by the semantic relationships found in the vocabularies which are known as (tags) in most of social media networks.**

This paper proposes a new model of semantic social media networks from the perspective of multi-agent systems. The multi-agent system is composed of two main functionalities: semantic indexing and tag ranking.

Index Terms—**Social Media, Semantic Social Ranking, Big Data, Multi-Agent Systems, Semantic Indexing, Tag Rank, LDA.**

I. Introduction

THE Social media are emerging field in information interchange, worldwide used and wanted. It is a challenging subject to do a research in social media field as it was and still affecting us in every aspect of our lives [1]. The improvement in retrieved contents in social media should be given attention as it reflects the quality and integrity of social media in general. The new perspective was to introduce semantics into social network to get Semantic Social Network in which relations and social graph will be composed according to the words meanings, especially keywords that are widely known as Tags in the social media network.

In current social media networks, Links between contents are constructed by many ranking techniques according to the way to deal with data and importance and priority of data. Such as posts in Facebook, hashtags in Twitter, Job and Experiences in LinkedIn,..Etc. and so data must be ranked in a way that links constructing the social graph will reflect natural distribution and connection between nodes of the social media. Rank of each node is given by making iterative process of weights in network. In Semantic Social Networks[2], this weight can be given according to semantic content of the social media node.

Semantic Content of Semantic Social Network, which is large and complex collections of data and that, is known nowadays as "***Big Data***" [3]must be indexed before ranking process. Introducing semantic indexing algorithms to process content of Semantic Social Networks can achieve this. Improving indexing output and choosing the proper ranking algorithm will affect the quality of the social graph and how nodes will be linked in semantic social network.

The existence of various ranking algorithms depending on how dealing with content which affects the quality of the output of the ranking. Therefore, the ranking of contents in social media should be based on some criteria that reflects related topics or links to the content. This can be achieved by depending on semantic indexing algorithms that gives the actual relations depending on the topic of the contents.

For Indexing and Ranking processes. The Concept of Multi-Agent system is a great addition to give good, improving, and self-learning mechanism especially in social networks. Multi-Agent Systems are computerized system composed of multiple interacting intelligent agents within an environment which can be used to solve problems [4].

An agent is a computer system that is capable of independent action on behalf of its user or owner [5] (figuring out what needs to be done to satisfy design objectives, rather than constantly being told).

To improve the output of ranking, improve the performance of indexing and ranking agents to get autonomous semantic social network linking and building of social graph. Improving indexing should be introduced by enhancement in algorithm to be applied in indexing process. Moreover, improving ranking must be met by semantic content analysis that makes the linking similar according to subjects or keywords on the social media content. In addition, processing time must be taken in consideration.

II. Ranking Algorithms

Ranking is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. The top popular algorithms used in social media are:

A. Page Rank (PR):

Commonly in Google, in Page Rank [9] if a page contains important links towards it then the links of this page towards the other page are also to be considered as important pages.

And the Page Rank considers the back link in deciding the rank score.
PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that websites that are more important are likely to receive more links from other websites. The Page Rank considers the back link in deciding the rank score. So assume we have two pages u and v:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)} \quad (1)$$

Where B_u is the set of all pages linking to page u. And $L(v)$ is the number of links from page v.
Considering damping factor the page rank will be:

$$PR(u) = (1 - d) + d \sum_{v \in B_u} \frac{PR(v)}{L(v)} \quad (2)$$

This rank algorithm does not reflect the content of pages but it concentrate on the number of links related to a page.

B. Weighted Page Rank (WPR):

Weighted page rank algorithm (WPR) [9] is the modification of the original page rank algorithm. WPR decides the rank score based on the popularity of the pages by taking into consideration the importance of both the in-links and out-links of the pages. This algorithm provides high value of rank to the more popular pages and does not equally divide the rank of a page among its out-link pages. Every out-link page is given a rank value based on its popularity. Popularity of a page is decided by observing its number of in and out links. For example for the pages u, p and v the weight rank is:

$$W^{in}_{(v,u)} = \frac{I_u}{\sum_{p \in R(v)} I_p} \quad (3)$$

Where I_u, I_p are numbers of in-links of pages u and p, $R(v)$ reference page list of page v

$$W^{out}_{(v,u)} = \frac{O_u}{\sum_{p \in R(v)} O_p} \quad (4)$$

Where O_u, O_p are numbers of out-links of pages u and p.

C. Edge Rank

Is the name commonly given to the algorithm that Facebook [10] uses to determine what articles should be displayed in a user's News Feed. Every action their friends take is a potential newsfeed story. Facebook calls these actions "Edges." That means whenever a friend posts a status update, comments on another status update, tags a photo, joins a fan page, or RSVP's to an event it generates an "Edge," and a story about that Edge might show up in the user's personal newsfeed.

It would be completely overwhelming if the newsfeed showed all of the possible stories from your friends. Therefore, Facebook created an algorithm to predict how interesting each story will be to each user. Facebook calls this algorithm "EdgeRank" because it ranks the edges. Then they filter each user's newsfeed to only show the top-ranked stories for that particular user. The general equation of this algorithm is:

$$Edge\ Rank \sum = u_e * w_e * d_e \quad (5)$$

Where u_e is the affinity score (between viewing users and edge creator). w_e *Weight for the edge type (create, comment, like, tag, etc.)* and d_e *time decay factor.*

D. Tag Rank

Tag Rank [11] is new suggested technique that is similar to page rank but it works on tags and links between nodes according to existence of tag in contents of social media.
This algorithm digs the annotation behavior of the web users, calculates the heat of the tags. By using time factor of the new data source tag and the annotations behavior of the web users. It can response the true quality of tags more externally and improve the veracity of page ranking. This algorithm provides a better authentication method for ranking the web pages. The results of this algorithm are very accurate and this algorithm index new information resources in a better way.
In Simple way, as we get the semantic index by indexing agent, the ranking agent job is to build Tag-Pair Weight Matrix (TWM) [12] as a rank matrix depending on the indexing result.
The mathematical model of the TagRank is:
First creating TFM which is (Tag Frequency Matrix) which is the sum of Tag Matrices TM depending on tag simultaneous appearance:

So $TM_{(i,j)} = 0$: $tag\ i\ and\ tag\ j$ do not appear simultaneously on certain content.

And $TM_{(i,j)} = 1$: $tag\ i\ and\ tag\ j$ appear simultaneously on certain content.

And so Tag Frequency Matrix is

$$TFM_{(i,f)} = \sum_{k=1}^{m} TM_k(i,f) \quad (6)$$

Lastly, the Tag-Pair Weight Matrix will be:

$$TWM_{(i,f)} = TSM_{(i,f)} \times TFM_{(i,f)} \quad (7)$$

Where $TSM_{(i,j)}$ is an entry of tag-pair similarity matrix.
Based on semantic social network perspective, we find that Tag Rank is the best option to go on with in our proposed model.

III. Indexing Algorithms

Indexing algorithms -generally in search engines- collects, parses, and stores data to facilitate fast and accurate information retrieval. Index design incorporates interdisciplinary concepts from linguistics, cognitive psychology, mathematics, informatics, and computer science. An alternate name for the process in the context of search engines designed to find web pages on the Internet is web indexing.
Popular engines focus on the full-text indexing of online, natural language documents. Media types such as video, audio and graphics are searchable.
To get the best result in indexing semantic indexing algorithms introduced to get actual index of contents of the content of social media.
Many Algorithms of semantic indexing were developed to reflect actual overview of semantic content of the documents, pages and other contents of social media. Such as:

A. TF-IDF (Term Frequency –Inverse Document Frequency)

It is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining [8]. The tf-idf value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. The mathematical equation for this technique is:

$$tf - idf_{t,d} = tf_{t,d} \times idf_t \quad (8)$$

Where $tf - idf_{t,d}$ is the score between query t and document d. $tf_{t,d}$ is the term frequency and idf_t is the Inverse Document Frequency.

Nowadays, tf-idf is one of the most popular term-weighting schemes. Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query

B. LSI (Latent Semantic Indexing)

It is another technique in natural language processing to discover information about the meaning behind words [14]. LSI analyzes relations between set of documents and terms they contain and assume that words that are close in meaning will occur in similar pieces of text. Then LSI constructs matrix of words (terms) per document, and using singular value decomposition to reduce and divide the big matrix into small orthogonal components. To finally represent vectors of words in documents.

C. PLSI (Probabilistic Latent Semantic Indexing)

Is a statistical technique for analysis of co-occurrence data [15]. Compared to standard latent semantic analysis which stems from linear algebra and downsizes the occurrence tables (usually via a singular value decomposition), PLSI is based on a mixture decomposition derived from a latent class model. Instead of matrices, PLSI uses probability methods to represent semantic. Instead of using matrices, (PLSI) uses a probabilistic method. Its graphical model is as

$$P(w|d) = P(d) \sum_c P(c|d)P(w|c) \quad (9)$$

Where d is the document index, c is word's topic drawn from $P(c|d)$, and w is word drawn from $P(w|c)$. And both $P(c|d)$ and $P(w|c)$ are modeled as multinomial distributions.

D. LDA (Latent Dirichlet Allocation)

is an improvement of PLSI by generalizing it using Dirichlet Prior as a variable reflects normal distribution of words in documents [16].

LDA is a mixture model. It assumes that each document contains various topics, and words in the document are generated from those topics. All documents contain a particular set of topics, but the proportion of each topic in each document is different.

The generative process of the LDA model can be described as follows:

1- Choose a multinomial distribution φ_z for each topic z from a Dirichlet distribution with parameter β
2- For each document d choose a multinomial distribution θ_d from a Dirichlet distribution with parameter α
3- For each word token w in document d Choose a topic $z \in \{1, \dots K\}$ from the multinomial distribution θ_d
4- Choose word w from the multinomial distribution $\emptyset_z$
5- Thus the likelihood of generating corpus is:

$$P(Doc_1, \dots, Doc_N|\alpha, \beta) = \iint \prod_{z=1}^{N} P(\varphi_z|\beta) \prod_{d=1}^{N} P(\theta_d|\alpha) \left(\prod_{i=1}^{N_d} \sum_{z_i=1}^{K} P(z_i|\theta)P(w_i|z_i\varphi)\right) d\theta d\varphi \quad (10)$$

As previous researches [17] compared between semantic indexing algorithms LDA was the best according to the quality of output, which can be measured by perplexity, log-likelihood, precision and recall.

In this paper, we will use this algorithm and we will find the best output of LDA indexing process based on modifying the parameters affecting the result (α, β and K).

IV. Proposed Model

A. System Architecture

Our proposed model consists of the following:

-Documents: which are sets of raw data from social networks to be processed.

-Indexer (Indexing Agent): In this part, three main processes are carried out; initializing documents, parsing document and indexing using semantic indexing algorithm.

-Index: it is the output of the document after indexer agent job completes. It contains the topic probabilities per document. And the word probability per topic. Topics are taken to be Tags for the first time. To be processed in ranking process.

-Ranking Agent: In this part, we get the probabilities of topics per document. Then process them to be ranked as tags. Using certain ranking algorithm.

-Social Graph: the output of ranking agent will be used to build links between social nodes. Figure1. Shows how this model is composed.

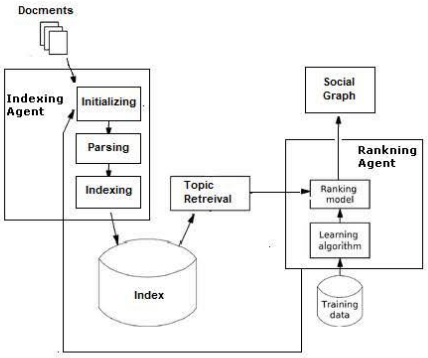


Figure1: System Architecture.

B. Algorithm

In the beginning, data in documents is collected and then parsed.

After that, Indexing Agent uses LDA algorithm starts to build the index of semantics, the result index that contains the topics and the most common word in each topic will be used to get the tags. Finally, Ranking Agent builds rank matrix of Tags to give the social graph link network. See the next Block for Algorithm1. MSSNT, which is abbreviation of (Multi-agent Semantic Social Network TagRank) algorithm that show how agents work.

Algorithm1. MSSNT
Input: Document Collocation Dataset
Start
//Indexing Agent{
Rule 1: Get Document
Rule 2: Parse Document Content
for i=1 to n **do**
Rule 3: Start LDA Indexing Algorithm
end for
Output Index $(\theta_{t_1}, \theta_{t_2}, \dots, \theta_{t_n})$
end } //end of indexing agent job

//Ranking Agent{
Start
for j=1 to n **do**
//repeat until all tags which have larger ranks than threshold τ
Repeat{
//select tag 1 and tag 2 which are columns and rows of Max$(\theta_{t_j}, \theta_{t_{j+1}})$
Select Max$(\theta_{t_j}, \theta_{t_{j+1}})$
Condition: While (Max$(\theta_{t_j}, \theta_{t_{j+1}}) \geq \tau$) {// τ is threshold
Select Max$(\phi_{w_j}, \phi_{w_{j+1}})$
}
j=j+1;
} // until (all tags which are larger than τ processed).
Build Links between Tags
end} //end of Ranking Agent job.
Output social graph

V. Simulation Results

In this section, we will do brief test on both algorithms to show how the system will work.

For LDA we used the Gibbs Sampling and its LDA model. The dataset is used was *psychreview* dataset. Which contains Psychology Review Abstracts and collocation Data.

Using Matlab, we tried to check what values of

We use perplexity and log-likelihood as the criteria to choose the three effective parameters in the LDA algorithm which are: α, β, and K.

Perplexity is a measurement of how well a probability distribution or probability model predicts a sample. It may be used to compare probability models. A low perplexity indicates the probability distribution is good at predicting the sample.

Log-likelihood is the natural logarithm of the likelihood function, called the log-likelihood. Likelihood function is a function of the parameters of a statistical model given data. Likelihood functions play a key role in statistical inference, especially methods of estimating a parameter from a set of statistics.

Log-likelihood is more convenient to work with. Because the logarithm is a monotonically increasing function, the logarithm of a function achieves its maximum value at the same points as the function itself, and hence the log-likelihood can be used in place of the likelihood in maximum likelihood estimation and related techniques.

Best choices of α and K have strong correlation, which chooses parameters as $\alpha = 50/K$, $\beta = 0.1$. For both datasets, we have the true label of every document. Therefore, an intuitive guess of K would be the number of classes. Grid search for all three parameters requires too much work, so we first fix K as the number of classes, and apply grid search of $\alpha$and β. After deciding best α and β for fixed K, we then search for best K with fixed β and changing α,under the assumption that $\alpha \propto 1/K$

In simulation we select $\alpha = \{\frac{0.01}{K}, \frac{0.1}{k}, .., \frac{1}{k}\}$
And $\beta = \{0.1, 1, 2\}$
And K={2, 3, 4, 5,6}

First, we fix the values of α and β to find the best value of K. The results was as shown on Figure-2.

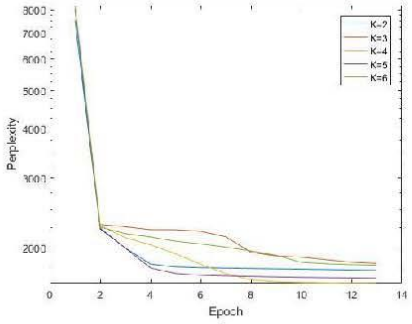


Figure2: Perplexity with changing Topic Classes number (K)

As shown in Figure2. When K=4 we have the minimum perplexity, which means when we classify collocation into 4 topic classes we get the best results in indexing.
With fixing K value to 4 and testing α and β we get the effective parameters where test results shows that $\alpha = 0.7/K$ and $\beta = 0.1$.
The Output of the Indexing process done by the Indexer Agent is used as input for Ranking Agent.

Using MATLAB tag rank is designed as finding matrix of maximum product between tags. And we set that links between tags must be when probability of tag in document is more than 0.4 as a threshold. The output for the first 50 documents of the collocation are as shown in Figure3.

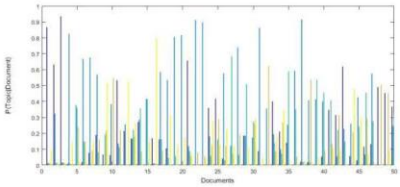


Figure3. Topic per Document distribution..

It is seen in Figure3. That the topics per document distribution is representing all the possibilities of linking documents to each other. However, after using TagRank to filter links according to specific threshold (τ =0.4) links will be as shown in Figure4. That means Tags links are constructed among this criterion.

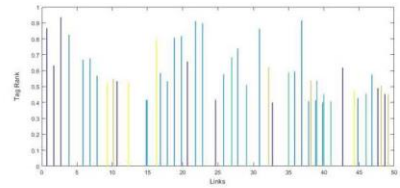


Figure4: Tag Rank Linking distribution

VI. Conclusion and Future Work

In this paper, we propose new model of social networks depending on semantics, with using semantic indexing methods and rank algorithms. In addition, show in test how this idea will be implemented.
In the Future, more modification on Tag rank algorithm needed. Also, further improvement of LDA to be observed.

References

[1] Obar, Jonathan A., Wildman, Steve. *Social media definition and the governance challenge: An introduction to the special issue.* Telecommunications policy. 39 (9): 745–750. doi:10.1016/j.telpol.2015.

[2] Stephen Downes. *The Semantic Social Network*. February 14, 2004.

[3] Boyd, Dana; Crawford, Kate. *Six Provocations for Big Data*. Social Science Research Network: A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society. (September 21, 2011). doi:10.2139/ssrn.1926431.

[4] Wooldridge, Michael. *An Introduction to MultiAgent Systems*. John Wiley & Sons. (2002) p. 366. ISBN 0-471-49691-X.

[5] Kubera, Yoann; Mathieu, Philippe; Picault, Sébastien. *Everything can be Agent*. Proceedings of the ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2010), Toronto, Canada: 1547–1548.

[6] Mathijs deWeerdt, Yingqian Zhang, Tomas Klos, *Multiagent Task Allocation in Social Networks*. Autonomous Agents and Multi-Agent Systems, 2012, Volume 25, Number 1, Page 46

[7] Wasserman, S., Faust, K. *Social network analysis: methods and applications*. Cambridge University Press, Cambridge (UK) (1994).

[8] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*. Cambridge University Press. 2008.

[9] S.Prabha, K.Duraiswamy,J.Indhumathi, *A Comparative Analysis of Different Page Ranking Algorithms*. International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:8, No:8, 2014

[10] http://edgerank.net/ accessed in September 1, 2017.

[11] S. Jie, C. Chen, Z. Hui, S. Rong-Shuang, Z. Yan and H. Kun. *TagRank: A New Rank Algorithm for Webpage Based on Social Web*. 2008 zTechnology, Singapore, 2008, pp. 254-258.

[12] DaeHoon Hwang, *Comparison and Evaluation of Highly Related Tag-Pairs Extraction Methods*. International Journal of Multimedia and Ubiquitous Engineering Vol.10, No.9 (2015).

[13] Dae-Hoon Hwang, *Design of Tag Ranking Algorithm Based on Cluster*. Advanced Science and Technology Letters Vol.133 (Information Technology and Computer Science 2016), pp.194-200.

[14] Scott Deerwester; Susan T Dumais; George W Furnas; Thomas K Landauer; Richard *Indexing by Latent Semantic Analysis* Journal of the American Society for Information Science (1986-1998); Sep 1990.

[15] Hofmann, Thomas *Probabilistic Latent Semantic Indexing*. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999.

[16] Blei, David M.; Andrew Y. Ng; Michael I. Jordan. *Latent Dirichlet Allocation* . Journal of Machine Learning Research. 3: 993–1022. doi:10.1162/jmlr. 2003.3.4-5.993.2003.

[17] Wang, Y., Lee, J.-S. and Choi, I.-C. *Indexing by Latent Dirichlet Allocation and an Ensemble Model*. Journal of the Association for Information Science and Technology, 67: 1736–1750. doi:10.1002/asi.23444. 2016.

Appendix B: Published Papers

B.2:

Second Paper:

Intelligent Social Networks Model Based On Semantic Tag Ranking

Accepted in the International Journal of Web & Semantic Technology (IJWesT) journal, and will be published on July 2018.

Intelligent Social Networks Model Based On Semantic Tag Ranking

Rushdi Hamamerh[1] and Sameh Awad[2]

[1]Department of Computer Engineering, Al-Quds University, Abu Dies, Palestine

rushdi@staff.alquds.edu

[2]Department of Information Technology, Birzeit University, Ramallah , Palestine

sfawad@birzeit.edu

Abstract

Social Networks has become one of the most popular platforms to allow users to communicate, and share their interests without being at the same geographical location. With the great and rapid growth of Social Media sites such as Facebook, LinkedIn, Twitter…etc. causes huge amount of user-generated content. Thus, the improvement in the information quality and integrity becomes a great challenge to all social media sites, which allows users to get the desired content or be linked to the best link relation using improved search / link technique. So introducing semantics to social networks will widen up the representation of the social networks.

In this paper, a new model of social networks based on semantic tag ranking is introduced. This model is based on the concept of multi-agent systems. In this proposed model the representation of social links will be extended by the semantic relationships found in the vocabularies which are known as (tags) in most of social networks. The proposed model for the social media engine is based on enhanced Latent Dirichlet Allocation(E-LDA) as a semantic indexing algorithm, combined with Tag Rank as social network ranking algorithm. The improvements on (E-LDA) phase is done by optimizing (LDA) algorithm using the optimal parameters. Then a filter is introduced to enhance the final indexing output. In ranking phase, using Tag Rank based on the indexing phase has improved the output of the ranking. Simulation results of the proposed model have shown improvements in indexing and ranking output.

Keywords

Social Network, Multi-Agent Systems, Semantic Indexing, Tag Rank, LDA, E-LDA.

1. INTRODUCTION

Social networks are emerging field in information interchange, worldwide used and wanted. It is a challenging subject to do a research in social media field as it was and still affecting us in every aspect of our lives [1].

Ellison and Boyd defined social networks (SN) as web-based services that allow users to build a public or semi-public profile within a system, connect to a list of other users by sharing a

connection, and view and extend their list of connections and those made by others within the system. The nature of these connections may vary from (SN) site to another [2].

In current social networks, Links between contents are constructed by many ranking techniques according to the way to deal with data, importance and priority of data. Such as posts in Facebook, hashtags in Twitter, Job and Experiences in LinkedIn, etc. and so data must be ranked in a way that links constructing the social graph will reflect natural distribution and connection between nodes of the social networks. Rank of each node is given by making iterative process of weights in network. In Semantic Social Networks, this weight can be given according to semantic content of the social network node.

Semantic Content of Social Network which is large and complex collections of data and that is known nowadays as "Big Data" [3] must be indexed before ranking process. This can be achieved by introducing semantic indexing algorithms to process content of Social Networks [4]. Improving indexing output and choosing the proper rank algorithm will affect the quality of the social graph and how nodes will be linked in social network.

2. MULTI-AGENT SYSTEMS

For Indexing and Ranking processes. The Concept of Multi-Agent system (MAS) is a great addition to give good, improving, and self-learning mechanism especially in social networks. Multi-Agent Systems are computerized system consisted of multiple agents that they interact intelligently within the environment which can be used to solve problems [5].

the agent precepts data and grabs the documents from the environment (SSN) and using the learning elements it updates the performance elements. And it builds the knowledge base by updating the learning elements based on critics that represents the feedback from the whole system that the agent is working in. and this knowledge base generates problems that can be used as condition rules to be used in the decision that the agent will make to do the needed action in the environment.

In social networks, the multi-agent implementation theories have two main perspectives: user perspective and network perspective.

In user perspective, the agents will be the user accounts [6], which means each account will act as agent in mediating data and negotiating connections with the other agents to enlarge their social network.

Nevertheless, in the network perspective, the agents will be carrying out some central operations such as filtering data, managing connectivity, and building the social graph.

Because of the semantics in social network is being discussed, the perspective of semantics must be an important role to be done by agents in the multi-agent implementation of social network.

In this paper, the concentration will be on some roles done by agents, which are parsing data, building semantic index of the data, then ranking this index, and finally build connections between contents according to the rank output.

3. SEMANTIC INDEXING - LATENT DIRICHLET ALLOCATION (LDA)

Indexing algorithms - mainly in search engines - collect, parses, analyse and store data to facilitate quick and accurate information retrieval [7]. Index design includes interdisciplinary concepts from linguistics, cognitive psychology, mathematics, computer science and informatics. An alternative name for the process in the context of search engines intended for searching web pages on the Internet is web indexing.

When dealing with information retrieval, stored documents are identified by sets of terms that are used to represent the contents of the document. The indexing process is the assignment of the index for documents in the collection of documents. The index of terms can be predefined as a fixed set of controlled vocabulary or can be any additional words that the indices consider to be related to the topic of the document.

One of the most popular indexing algorithms is Latent Dirichlet Allocation (LDA) [8], is a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modelled as a finite mixture over an underlying set of topics. Each topic is, in turn, modelled as an infinite mixture over an underlying set of topic probabilities. LDA assumes that each document contains different topics, and words in the document are generated from these topics. All documents contain a specific set of topics, but the proportion of each topic in each document is different. The generative process of the LDA model can be described as follows [9]. Assuming document w in a corpus D:

1- Choose N ~ Poisson (ξ).

2- Choose θ ~ Dir(a).

3- For each of the N words w_n:

Choose a topic z_n ~ multinomial distribution (θ)

Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Many simplifying assumptions are made in this basic model, such as removing some subsequent sections.

First, the dimensionality k of the Dirichlet distribution which means the dimensionality of the topic variable z is assumed known and fixed. Second, the word probabilities are parameterized by k $k \times V$ matrix β where $\beta_{ij} = p(w^j = 1 \,|z^i = 1)$ which for now is treated as a fixed quantity that is to be estimated. Finally, the Poisson assumption is not critical to anything that follows and more realistic document length distributions can be used as needed.

Furthermore, note that N is independent of all the other data generating variables (θ and z). It is thus an ancillary variable and its randomness will generally be ignored in the subsequent development. A k-dimensional Dirichlet random variable θ can take values in the (k−1)-simplex (a k-vector θ lies in the (k−1)-simplex if $\theta_i \geq 0, \sum_{i=1}^{k} \theta_i = 1$), and has the following probability density on this simplex:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \dots \theta_k^{\alpha_k - 1} \qquad (1)$$

Where the parameter α is a k-vector with components α_i >0, and where $\Gamma(x)$ is the Gamma function. The Dirichlet is a convenient distribution on the simplex—it is in the exponential family, has finite dimensional sufficient statistics, and is conjugate to the multinomial distribution. Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w is given by:

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta) \quad (2)$$

Where $p(z_n|\theta)$ is simply θ_i for the unique i such that$z_n^i = 1$. Integrating over θ and summing over z, then the marginal distribution of a document will be:

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta)p(w_n|z_n, \beta)\right) d\theta \quad (3)$$

Finally, the probability (or the log-likelihood) of generating corpus is:

$$p(D|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta)p(w_{dn}|z_{dn}, \beta)\right) d\theta_d \quad (4)$$

4. PROPOSED MODEL

The proposed model for semantic tag ranking for social network is based on enhanced LDA. The input in this model is the document collection where it contains the word per document count. Then the final output will be the ranking of the tags, which are the Tag Rank results of the topics index. The proposed model is based on two main phases: the indexing phase which is carried out by the indexing agent, and the ranking phase which is carried out by the ranking agent. In indexing phase, the input is the document collection where it contains word and document count. In this phase, the initialization is done then document parsed to get the initial index to be processed by (LDA) algorithm. The output of this phase is the semantic index, which contains word-per-topic distribution and topic-per-document distribution.

In this proposed model, the focus on the topic-per-document to be processed as tags. Therefore, in the next phase, which is the ranking phase the input will be the topic-per-document distribution that came as index matrix. In ranking phase, the input will be processed by Tag Rank algorithm with the help. The final output will the Tag ranking matrix that will be sent to build the social links in the semantic social network. Figure1. Shows the proposed model architecture.

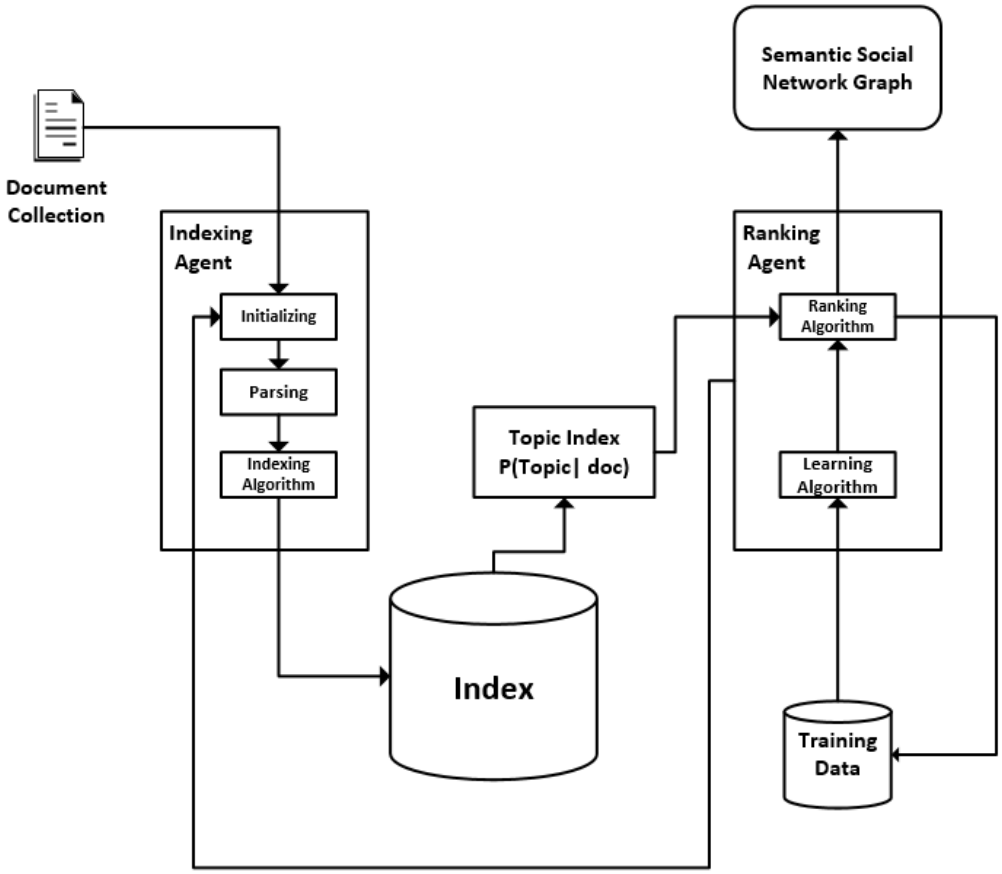


Figure2: System Architecture.

The indexing phase has seven sequential steps to build the topic per document index document based on the document collection to be processed. Figure2. Shows the steps of this phase:

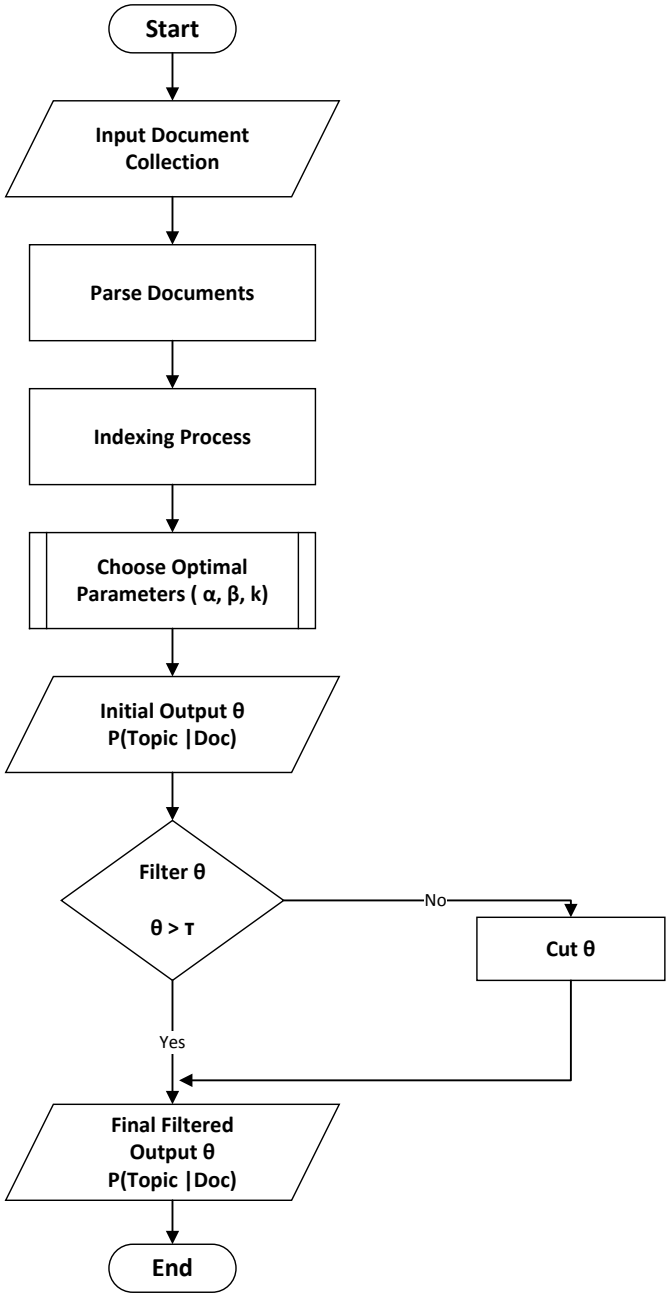


Figure2: Flowchart of Indexing Phase.

The start is with the input of document collection, which is parsed then indexed with choosing the optimal parameters (a, β and K) which increases the precision and recall of the output.

Then the output will be probability of topic per document that will be filtered by specific threshold (τ) that will be chosen by experiment in the simulation. The final output will be the filtered (θ)which is the output of the enhanced LDA algorithm. Which is called E-LDA.

The next phase is ranking phase. It starts with the output of E-LDA algorithm with checking that (θ) is higher than the threshold (τ). Then the Tag Rank algorithm start to rank (θ) as initial tag rank. The ranking algorithm is simply here to maximize the rank. Each document will get the higher topic ranking to be the first tag. In addition, documents will be descending ranked for each tag i.e. for each topic. Figure3. Shows the steps of ranking phase.

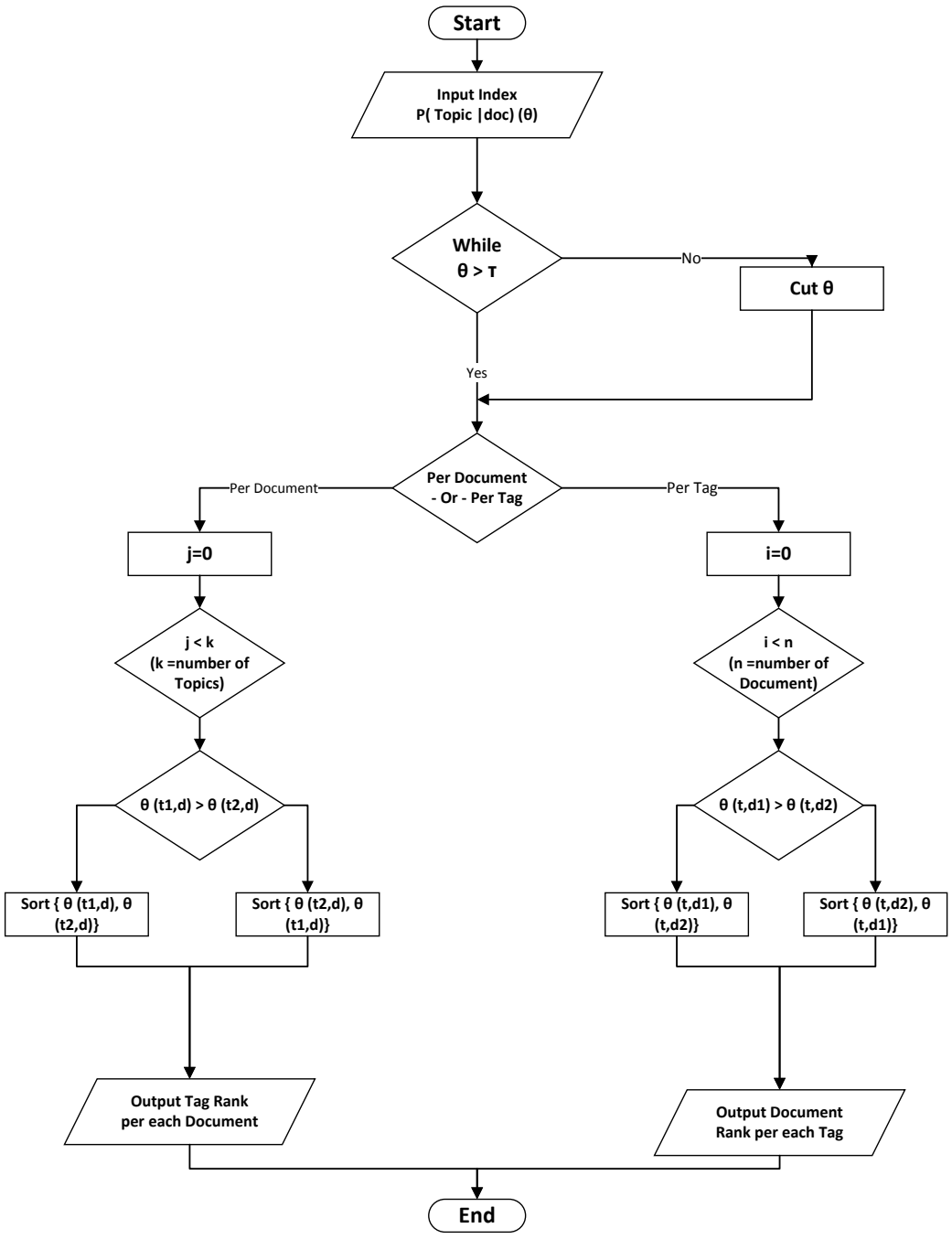


Figure3: Flowchart of Ranking Phase

As shown in these flowcharts it is obvious that there are two main intelligent agents that are carrying out the system functions. Indexing and ranking agents. The next pseudocode shows the steps of the algorithm of the semantic tag ranking.

Algorithm 1. Intelligent Semantic Tag Ranking

Input: Document Collection

Start

//Indexing Agent{

Rule 1: Get Document

Rule 2: Parse Document Content

for *i=1* to *n* do //n= number of document records

Rule 3: Start LDA Indexing Algorithm

end for

Rule 4: Filter

{

for *i=1* to *n* do //n= number of document records

Select $\theta_{t_i} where\ \theta_{t_i} > \tau$ // τ is threshold

end for

}

Output Index ($\theta_{t_1}, \theta_{t_2}, \dots, \theta_{t_n}$)

end } //end of indexing agent job

Input: Index ($\theta_{t_1}, \theta_{t_2}, \dots, \theta_{t_n}$)

//Ranking Agent{

Start

for *i*=1 to *n* do //n= number of tags

//repeat until all tags which have larger ranks than threshold τ

Repeat{

//select document 1 and document 2 to be compared and maximized

Select Max(θ_i, θ_{i+1})

Condition: While (Max(θ_i, θ_{i+1}) $\geq \tau$) { // τ is threshold

Select Max(θ_i, θ_{i+1})

Sort (θ_i, θ_{i+1})

}

***i*=*i*+1**

} // until (all tags which are larger than τ are processed).

for *j*=1 to *k* do //k= number of documents

//repeat until all documents which have larger ranks than threshold τ

Repeat{

//select tag 1 and tag 2 which are columns and rows of Max$(\theta_{t_j}, \theta_{t_{j+1}})$

Select Max$(\theta_{t_j}, \theta_{t_{j+1}})$

Condition: While (Max$(\theta_{t_j}, \theta_{t_{j+1}}) \geq \tau$) { // τ is threshold

Select Max$(\theta_{w_j}, \theta_{w_{j+1}})$

Sort $(\theta_{w_j}, \theta_{w_{j+1}})$

}

***j*=*j*+1**

} // until (all tags which are larger than τ are processed).

Build Links between Tags

Output Tag Rank records

end } //end of Ranking Agent job.

5. SIMULATION RESULTS

This section presents the simulation experiment for the proposed model. The concept of combining index resulting from LDA with threshold applied to be as the Tag input for the ranking algorithm has to be proven by results and providing a good comparison between the proposed model phases, in both indexing and ranking phases

Simulation was carried out using MATLAB R2016a simulation software under Microsoft Windows 10 operating system.

The hardware platform that carried out the software is Intel core i7-3520M processor with 8 Gigabyte random access memory.

The simulation on the indexing phase will be carried out based on previous simulation works done by The Natural Language Processing Group at Stanford University [10], also on natural language labs on Iowa State University [11], and the research toolbox from University of California, Irvine [12], using their MATLAB functions to implement the enhanced LDA function.

The dataset is used was *psychreview* dataset. Which contains Psychology Review Abstracts and collocation Data. This dataset contains about 85000 records of words and documents. With the initial count of words for each document and the topic.

To evaluate the simulation, main four metrics were introduced; two for evaluating indexing which are precision and recall. The other two is for evaluating ranking and these metrics are mean average precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) [13].

Precision: is the ratio of the number of relevant documents retrieved to the total number of irrelevant and relevant documents retrieved:

$$\boldsymbol{Precision} = \frac{|\boldsymbol{relevant\ documents} \cap \boldsymbol{retreived\ documents}|}{|\boldsymbol{retreived\ document}|} \quad (5)$$

Recall: is the ratio of the number of relevant documents retrieved to the total number of relevant documents in the dataset:

$$\boldsymbol{Recall} = \frac{|\boldsymbol{relevant\ documents} \cap \boldsymbol{retreived\ documents}|}{|\boldsymbol{relevant\ document}|} \quad (6)$$

Mean Average Precision (MAP): is the precision-at-k score of a ranking y, averaged over all the positions k of relevant documents:

$$\boldsymbol{MAP} = \frac{\sum_{q=1}^{\boldsymbol{Q}} \boldsymbol{AveP}(\boldsymbol{q})}{\boldsymbol{Q}} \quad (7)$$

Where:

$$AveP = Average\ Precison = \frac{\sum_{k=1}^{n}(P(k) \times rel\,(k))}{number\ of\ relevant\ documents} \quad (8)$$

Q is the number of queries, and:

$$rel(k) = \begin{cases} 1, & when\ item\ at\ rank\ (k) is\ relevant \\ 0 \end{cases} \quad (9)$$

Normalized Discounted Cumulative Gain (NDCG): is a normalization of the Discounted Cumulative Gain (DCG) where (DCG) is a weighted sum of the relevancy degree of the ranked items:

$$\boldsymbol{NDCG_p} = \frac{\boldsymbol{DCG_p}}{\boldsymbol{IDCG_p}} \quad (10)$$

Where: $DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2\, i}$. (11)

And $\mathrm{IDCG_p}$ is the ideal DCG at position p

Based on previous research done earlier [14], the chosen the optimal parameters for LDA algorithm(α,β, and k), were $k = 4$, $\alpha = \frac{0.7}{k}$, And $\beta = 0.1$.

The next enhancement is to choose the best threshold (τ) to filter the output of the indexing process. Therefore, for the index output with the parameters that have been chosen before, calculating the precision and recall of the output with applying the filter. The result was as shown in Figure4.

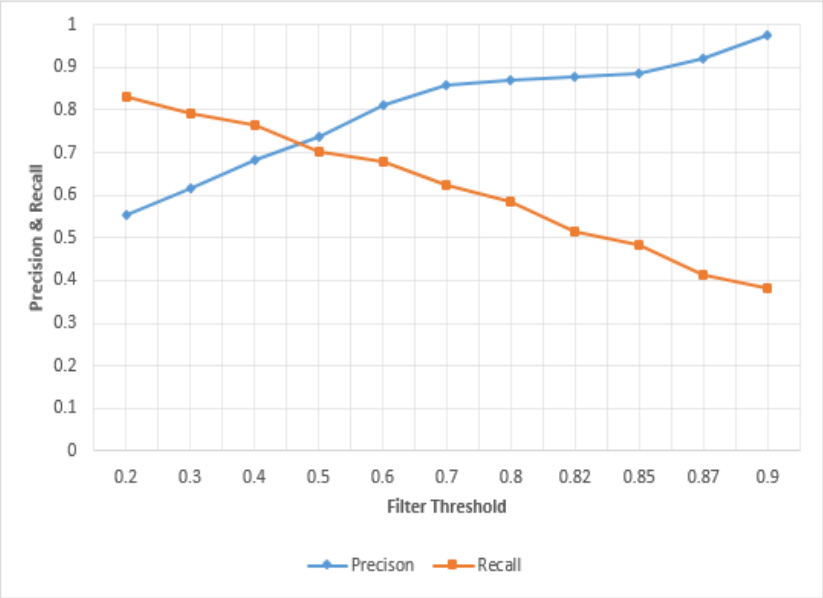
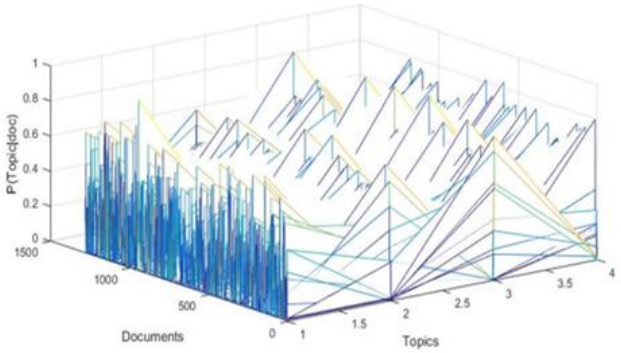
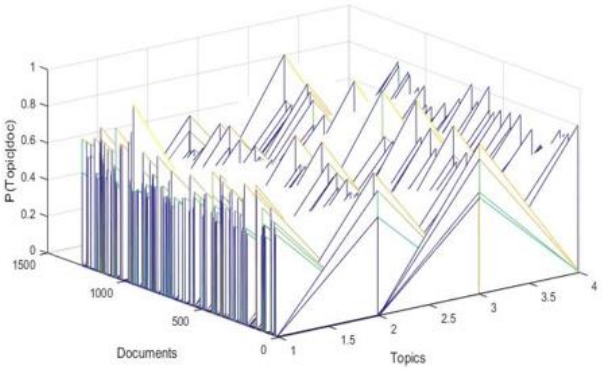


Figure4. Precision and Recall according to (τ)

It was noticed that the best combination of precision and recall is around τ=0.5. And so it is a good suggestion to choose this value as the threshold of the filter. The resulting algorithm with these enhancement is called “Enhanced LDA” abbreviated (E-LDA). Figure5. Shows how topic distribution is enhanced using this filter:



Before Filter



After Filter

Figure5. Topic Distribution in Document Collection according to the Filter.

The simulation for the indexing agent was carried out based on previous researches comparing indexing algorithms [9] [15]. Figure6 shows a comparison between (E-LDA) and these algorithms.

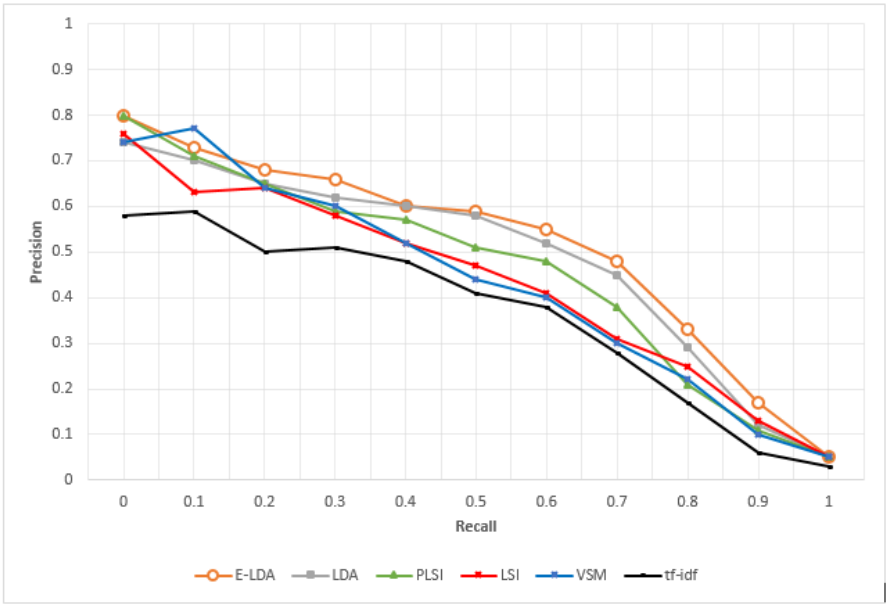


Figure6. (E-LDA) vs semantic indexing algorithms

As shown in Figure. E-LDA is enhanced from LDA with (4%). E-LDA has better precision vs. recall combination which means better relevancy in index output.

After indexing phase enhancement done, it is possible to combine tag rank with the output to get the semantic tag rank. Figure7. Shows the improvement in precision and recall between E-LDA and the Tag Rank.

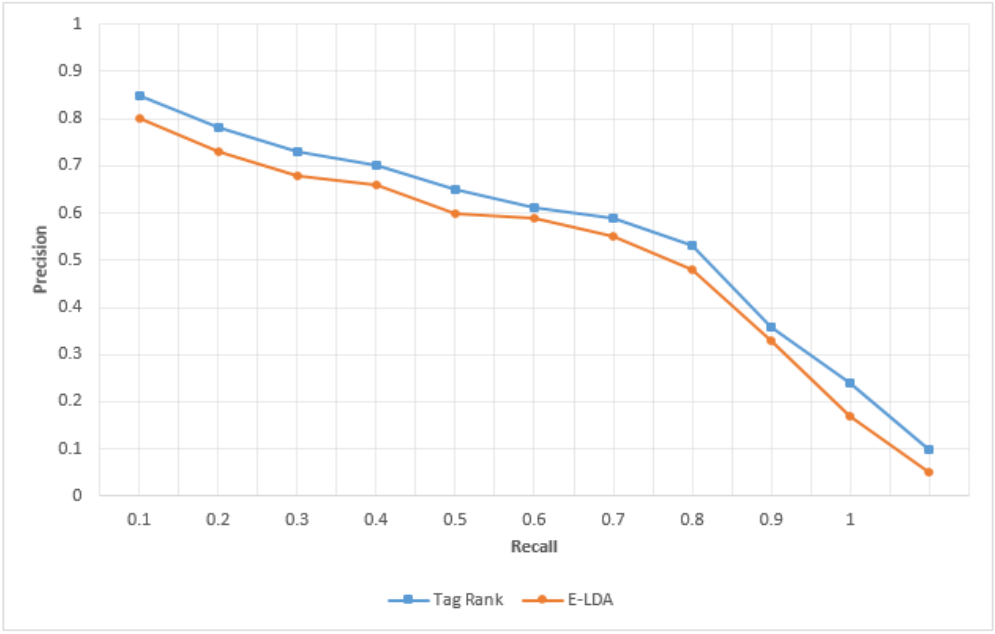


Figure7.Tag Rank vs. (E-LDA).

As shown in Figure7. Tag Rank shows better precision and recall than input from E-LDA with almost (5%).

Comparing Tag Rank with Page Rank (PR), Weighted Page Rank (WPR), Hyper-link Induced Topic Search (HITS) and Time Rank (TSPR) [16] [17] [18] [19]. And according to MAP and NDCG@(k=4) as (k=4) is the best parameter for the indexing algorithm LDA that was concluded earlier [14]. Figure8. Shows the comparison between ranking algorithms:

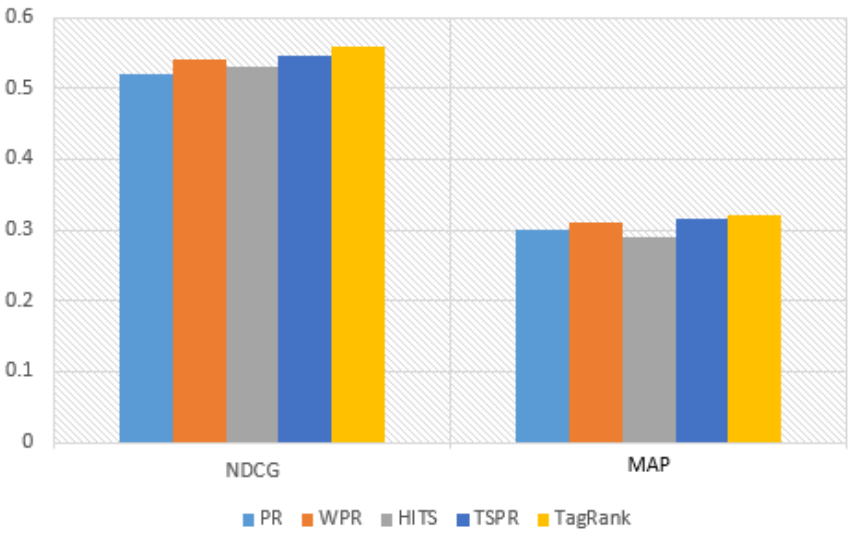


Figure8. Semantic Tag Rank vs. Ranking Algorithms

As shown in Figure8. Tag Rank shows the best MAP and NDCG values and so it could be said that Tag Rank is the best suitable ranking algorithm for this proposed model.

6. CONCLUSION AND FUTURE WORK

In this paper, the main aim is to provide new model of Social Network that is based on Multi-Agent Systems concept and the concept of semantic social network. This proposed model mainly consisted of two main agents: indexing agent that carries out enhanced Latent Dirichlet Allocation algorithm (E-LDA), and ranking agent that carries out Tag Rank algorithm. Enhanced LDA (E-LDA) is distinguished from other preceding indexing algorithms and simulation results show an increase precision and recall using E-LDA. E-LDA is enhanced from LDA with (4%), and shows better performance than other semantic indexing algorithms.

Semantic Tag Rank is also distinguished from other ranking agents as it deals with tags that is more relevant to social networks and also more relevant to semantics.

In the future, the term per topic index is suggested to be entered as tags to be processed by ranking agent. This means that we will have larger data to be ranked. So the processing conditions must be taken care of while implementing the system.

A new model of social networks depending on semantics is proposed, with using semantic indexing methods and rank algorithms. In addition, show in test how this idea will be implemented. Then building and implementing the proposed model to a semantic social network can be suggested. Either in an existing social network, or in new semantic social network programmed from the beginning based on the proposed model in this paper.

REFERENCES

[1] Obar, Jonathan A., Wildman, Steve. Social media definition and the governance challenge: An introduction to the special issue. Telecommunications policy. 39 (9): 745–750. doi:10.1016/j.telpol.2015.

[2] Boyd, Dana. Ellison, Nicole. Social Network Sites: Definition, History, and Scholarship, Michigan State University, (2007).

[3] Boyd, Dana; Crawford, Kate. Six Provocations for Big Data. Social Science Research Network: A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society. (September 21, 2011). doi:10.2139/ssrn.1926431.

[4] Stephen Downes. The Semantic Social Network. February 14, 2004.

[5] Wooldridge, Michael. An Introduction to MultiAgent Systems. John Wiley & Sons. (2002) p. 366. ISBN 0-471-49691-X.

[6] Franchi, Enrico , "A Multi-Agent Implementation of Social Networks", Proceedings of the 11th WOA 2010 Workshop, Dagli Oggetti Agli Agenti, Rimini, Italy, September 5-7, 2010.

[7] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, "Introduction to Information Retrieval". Cambridge University Press. 2008.

[8] Blei, David M.; Andrew Y. Ng; Michael I. Jordan. “Latent Dirichlet Allocation”. Journal of Machine Learning Research. 3: 993–1022. doi:10.1162/jmlr. 2003.3.4-5.993.2003.
[9] Wang, Y., Lee, J.-S. and Choi, I.-C. “Indexing by Latent Dirichlet Allocation and an Ensemble Model”. Journal of the Association for Information Science and Technology, 67: 1736–1750. doi:10.1002/asi.23444. 2016.
[10] The Natural Language Processing Group at Stanford University, https://nlp.stanford.edu/, accessed in October 1, 2017.
[11] Iowa State University, http://home.eng.iastate.edu, accessed in October 10, 2017.
[12] Matlab Topic Modeling Research Toolbox in University of California, Irvine, http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm, accessed in October 12, 2017.
[13] Järvelin, Kalervo and Jaana Kekäläinen. “IR evaluation methods for retrieving highly relevant documents.” SIGIR Forum 51 (2000): 243-250.
[14] R. Hamamreh and S. Awad, "Tag Ranking Multi-Agent Semantic Social Networks ," *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, 2017.
[15] Choi, In-Chan & Lee, Jaesung. “Document Indexing by Latent Dirichlet Allocation”. In proceedings of the 2010 international conference on data mining, At Los Angeles, 2010.
[16] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web”, Technical Report, Stanford Digital Libraries SIDL-WP-1999-0120, 1999.
[17] Wenpu Xing and Ali Ghorbani, “Weighted PageRank Algorithm”, In proceedings of the 2rd Annual Conference on Communication Networks & Services Research, PP. 305-314, 2004.
[18] Jon Kleinberg, “Authoritative Sources in a Hyperlinked Environment”, In Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 1998.
[19] H Jiang et al., "TIMERANK: A Method of Improving Ranking Scores by Visited Time", In proceedings of the Seventh International

Authors

[1] Rushdi A. Hamamreh has PH.D. in Distributed Systems and Networks Security, He graduated at the Saint Petersburg State Technical University in 2002; He is Associate Professor and Head of Computer Engineering at Al-Quds University. His research interests include Networks Security, Routing Protocols, Multiagent Systems and, Cloud and Mobile Computing.

[2] Sameh Awad graduated in Computer Engineering in 2008 from Al-Quds University. Since that he has been working in the Department of Information Technology in Birzeit University. In 2018 he has completed a MSc in Electronics and Computer Engineering from Al-Quds University.

Appendix C: Some MATLAB Codes

LDA.m

```
function [Phi,Theta,LL,LLAll]=LDA (WS,DS, T, Alpha, Beta, Iter, BURNIN, LAG,Log)

 % INPUT:
% WS: Word index, DS: Doc index K: number of topics
% Alpha : The prior for P(toic|doc) symmetric
% Beta : The prior for P(word|topic) symmetric
% Iter: number of iterations
% BURIN: burn-in period
% LAG: sampling lag
% OUTPUT:
% Phi: P(word|topic)
% Theta: P(toic|doc)
% LLAll: Log likelihood in each iterations
% LL: LogLikelihood


if nargin<9
   Log=0;
end
%% Initialization
if Log==1; display('Initialization...'); end
M=max(DS); V = max(WS);
NWZ = Beta*ones(V,T); NZM = Alpha*ones(M,T);
NZ = sum(NWZ); Z = zeros(length(WS),1);
 for w=1:length(WS)
       Z(w) = find(mnrnd(1,ones(1,T)/T )==1); % draw topic for each word
       NZM(DS(w),Z(w)) = NZM(DS(w),Z(w)) + 1;
       NWZ(WS(w),Z(w)) = NWZ(WS(w),Z(w)) + 1;
       NZ(Z(w)) = NZ(Z(w)) + 1;
 end
if Log==1; display('Done!'); end
```

```
%% Sampling...

Phi = zeros(V,T); Theta = zeros(M,T);
Phi2 = zeros(V,T); Theta2 = zeros(M,T);
LL=0;
SampleNum = 0;
LLAll = zeros(1,Iter);
for i = 1:Iter
if Log==1; display(sprintf('Processing %d of %d',i,Iter)); end
   for w=1:length(WS)
         % decrease three counts
         NZM(DS(w),Z(w)) = NZM(DS(w),Z(w)) - 1;
         NWZ(WS(w),Z(w)) = NWZ(WS(w),Z(w)) - 1;
         NZ(Z(w)) = NZ(Z(w)) -1;
         % update the posterior distribution of z, p(z_i)
         p = (NWZ(WS(w),:)./NZ).*NZM(DS(w),:);
         Z(w) = find(mnrnd(1,p/sum(p))==1);
         % increase three counts
         NZM(DS(w),Z(w)) = NZM(DS(w),Z(w)) + 1;
         NWZ(WS(w),Z(w)) = NWZ(WS(w),Z(w)) + 1;
         NZ(Z(w)) = NZ(Z(w)) + 1;
   end
   % log Likelihood
   LLAll(i)=log_multinomial_beta(NWZ)-
log_multinomial_beta(ones(V,T)*Beta);

   % Get Sample
   if i >= BURNIN || mod(i,LAG) == 0
   SampleNum = SampleNum + 1;
   Phi = Phi + bsxfun(@rdivide,NWZ,NZ);
   Theta= Theta + bsxfun(@rdivide,NZM,sum(NZM,2));
   LL=LL+LLAll(i);


   end
```

```
    %Applying Filter on Theta

    Theta2=Theta;

    Theta2(Theta<0.5)=0;

    Theta=Theta2;


end

if Log==1; display('Done!');
end



%% Get averaged Phi and Theta
Phi = Phi/SampleNum;
Theta = Theta/SampleNum;
LL = LL/SampleNum;
```

TagRank.m

```
function(tag, rank) = TagRank ( Theta)

n = 'enter a value for number of tags';

% entering the number of tags to compare per document

k='enter a value for number of documents ';

% entering the number of documents to compare per tag

Theta=[];
```

```
% get the filtered topic index

tou=0.5;

% define the filter threshold value (τ)

for i=1:1:n; % compare the tags per each document

  C=max(Theta(i),Theta(i+1));

while(max(Theta(i),Theta(i+1))>tou)

      C=max(Theta(i),Theta(i+1));

      theta=[Theta;C];

    B = sort(C) % output of the rank algorithm

end

end

for j=1:1:k; % compare the documents per each tag

  D=max(Theta(j),Theta(j+1));

while(max(Theta(j),Theta(j+1))>tou)

      D=max(Theta(j),Theta(j+1));

           theta1=[Theta;D];
```

```
    Z = sort(D) % output of the rank algorithm
end
end
% end of Tag Rank
end
```

النظم المتعددة الوكلاء في الشبكات الاجتماعية الدلالية بالاعتماد على تصنيف العلامة الدلالية

إعداد: سامح عبد الفتاح حسين عوض

إشراف: د. رشدي حمامرة

الملخص

أصبحت شبكات التواصل الاجتماعي واحدة من المنصات الأكثر شعبية والتي تتيح للمستخدمين التواصل، وتبادل مصالحهم دون أن يكونوا في نفس الموقع الجغرافي .مع النمو الكبير والسريع لشبكات التواصل الاجتماعي مثل Facebook، LinkedIn، Twitter ... الخ. تنتج كمية هائلة من المحتوى الذي ينشئه المستخدمون. وبالتالي، فإن تحسين جودة المعلومات ومصداقيتها أصبح تحدياً كبيراً لجميع شبكات التواصل الاجتماعي، والذي يسمح للمستخدمين الحصول على المحتوى المطلوب أو أن يتم ربطهم بناء على أفضل صلة باستخدام تحسين تقنية البحث والارتباط. لذا فإن إدخال الدلالات على الشبكات الإعلامية بما يعرف بالشبكات الاجتماعية الدلالية سيوسع تمثيل شبكات التواصل الاجتماعي.

تمثيل الشبكات الاجتماعية الدلالية للروابط الاجتماعية سيتم توسيعه من خلال العلاقات الدلالية الموجودة في المفردات التي تعرف باسم العلامات الدلالية (Tags) في معظم شبكات التواصل الاجتماعي.

يمكن ربط محتويات الشبكات الاجتماعية الدلالية باستخدام الوكلاء (Agents)، والذين يؤدون مهام محددة لجعل عملية الربط آلية، وذاتية التعلم وذكية. بالتالي تم طرح مفهوم النظم المتعددة الوكلاء لهذا البحث.

في هذه الأطروحة، اقترحنا نموذجا لشبكات التواصل الاجتماعية الدلالية من منظور الأنظمة متعددة الوكلاء (MSSNT). في هذا النموذج، يتكون النظام متعدد الوكلاء من وظيفتين رئيسيتين: الفهرسة الدلالية(Semantic Indexing) وتصنيف العلامات الدلالية (Tag ranking).

النموذج المقترح هو تحسين لناتج التصنيف، ولتحقيق ذلك ينبغي استخدام نوع من المرشحات(Filters) لزيادة رتبة المحتوى، وتحسين الرتبة يجب أن يتحقق من خلال تحليل المحتوى الدلالي الذي يجعل الربط في شبكات التواصل الاجتماعي وفقاً للمواضيع المتماثلة أو الكلمات الرئيسية الموجودة في محتوى تلك الشبكات.

النموذج المقترح لمحرك شبكات التواصل الاجتماعي يستند بشكل أساسي على خوارزمية توزيع "ديريكتليت" للمضامين (Latent Dirichlet Allocation) كخوارزمية الفهرسة الدلالية، إضافة إلى تصنيف العلامات الدلالية (Tag Rank) كخوارزمية التصنيف لشبكات التواصل الاجتماعي.

أظهرت نتائج المحاكاة أداءً أفضل في مرحلتي الفهرسة والترتيب. ففي مرحلة الفهرسة، تقدم خوارزمية E-LDA دقة أفضل (Precision) واسترجاع أفضل (Recall) بنسبة 4٪ من خوارزمية LDA غير المعدّلة، وأفضل أداء على الإطلاق مقارنة مع خوارزميات الفهرسة السابقة المستخدمة في شبكات الانترنت.

أما في مرحلة الترتيب، أظهرت خوارزمية ترتيب العلامات استنادًا إلى الموضوع في توزيع المستندات الناتج عن E-LDA أداءً أفضل في الدقة وتذكر بنسبة 5٪ تقريبًا. وأفضل النتائج حسب متوسط الدقة (MAP) ومعدل الربح التراكمي المخصوم (NDCG) مقارنة بخوارزميات الترتيب الأخرى.