1-1-2012

Fuzzy Thesauri Recommendation System For Web 2.0 Social networks

Touhid Ghasemi
Ryerson University

FUZZY THESAURI RECOMMENDATION SYSTEM FOR WEB 2.0 SOCIAL NETWORKS

by

Touhid Ghasemi

Bachelor of Science, Eastern Mediterranean University, Cyprus, 2007

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada, 2012

Author's Declaration

FUZZY THESAURI RECOMMENDATION SYSTEM FOR WEB 2.0 SOCIAL NETWORKS

Master of Science 2012

Touhid Ghasemi

Computer Science

Ryerson University

Abstract

In the information age with billions of documents available on the Internet, searching among these documents has become quite a challenge for researchers. Since most of the search methods are based on terms within the documents, identifying the relationship between the terms has always been important in the field of Information Retrieval. Using term relations in query expansion techniques is one of the most commonly used and successful approaches that are being used in order to help users find what they need. In this study a fuzzy set based methodology is exploited for the retrieval and analysis of data available in Web2.0 social networking sites. The documents in each server or node are used for building a knowledge-base that will be employed by the Recommendation System in order to provide domain specific suggestions, based on the friendship network in social networking sites. The results of the study show that the proposed methodology is reasonably scalable and can be employed on social networking sites.

Acknowledgements

My utmost gratitude to my supervisor, Dr. Abdolreza Abhari for his supervision and guidance, continuous support and encouragement. His advice were always inspiring. Thank you for provided me with constructive comments and timely support during my thesis write up.

I would also like to express my sincere gratitude to the members of my exam committee panel for their time and patient to read through my thesis and provide me with their valuable comments for better improvement of this work.

Special thanks goes to Dr. Hooman Tahayori for his intelligent advice, helpful suggestions, valuable insight and scientific input on Fuzzy logic part of the thesis.

I would also like to thank all the staff of the Department of Computer Science at Ryerson University for the guidance and technical support they provided me during the past years.

Thanks to my colleagues Abdolkarim Dabbagh, Kayvan Tirdad, Ahmad Gholizadeh, Nima Sharafeddin, Saeed Rashwand and Shahin Talaei from whom I have learned great things and not only in academic aspects. My special thanks to Negar Zohouri Haghian for her encouragement and enthusiastic support.

It is a pleasure to thank all those whom have helped and inspired me during my graduate study at Ryerson University. Those whom in one way or another contributed their feedback and assistance in completion of this study.

Last but not least, I would like to take this opportunity to thank all my family especially my parents, whom always been patient with me and supported me abroad in Cyprus and Canada to complete my studies.

Spring 2012

Dedication

To my parents.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Motivations and Objectives

1.1 Overview

The emergence of Internet and later World Wide Web (WWW) has allowed users to not only access large amounts of data including text, images, videos and other types of multimedia efficiently, but also store their information on the web. This never has been possible without the technological advancement. Individuals, businesses and governments heavily rely on Internet for faster retrieval, storage and management of their information of interest [1].

Advancements in the field of computer technology has changed the way we live and communicate with one another. Individuals are more dependent on Internet to obtain and exchange information as they have ever been before. Information are now stored digitally as compared to traditional paper-based methods. Individuals have switched from classically searching for information by referring to large number of books or asking human experts. The initial idea of storing information using computers was introduced by Vannevar Bush in his 1945 article titled "As We May Think" which was quite remarkable and as Singhal calls it quite "groundbreaking" [2]. It was the first time that someone has introduced the idea that large amounts of information stored could be automatically accessed.

As files and information were stored online, the need to search through them became a challenge for computer scientists. Archie [3] was the first tool created to search the Internet. It was developed by three students at McGill University in 1990 and later became commercialized. Archie could search the path or filenames but not the content of public anonymous FTP achieves. Soon after, other developments allowed the search of the titles and menu information. In 1993, the very first search engine − W3Catolog − was developed. W3Catalog [4] mirrored the sources available on the Web and changed the formats of the content into single entries which were searchable via a front-end. Today, the most popular search engines are Google with 80.76% market share, Yahoo! by 6.74% market share, Baidu by 6.06%, Bing by 4.27%, Ask by 0.55% and AOL by 0.36% market share among Internet users [5].

The emergence of Web 2.0, the next generation of World Wide Web, introduced a new era and a new perspective in the world of computer technology. Information sharing and user involvement in application designs are two of the examples that migration to Web 2.0 era has enabled us to do. Ever since the generation of Web 2.0, users are able to contribute in the content of Web. Wiki websites are the best example of the movement from the classical read only World Wide Web to Web 2.0. The very first wiki application called "WikiWikiWeb" was developed by Ward Cunningham and lunched in 1995 [6]. Wikis encourage user involvement by adding, removing or editing pages of information through the Wiki application that can be accessed by web browsers. It also allows creating links from technical terms or meaningful topics mentioned in one page to their associated page or article. Linkage between pages of information by associating the technical terms used in one page to another page that completely describes the term in full. A good example of wiki services is Wikipedia where over 21 million of articles are connected through an unstructured graph [7]. Blog publishing services such as Blogger and social networking services such as Facebook are other examples of this new trend. Thus the volume of information available on the Internet has dramatically increased.

The exponential growth of data stored on the Internet has introduced a series of challenges in the field of information retrieval. Mainly, search engines take a key word or a string of information and find the exact match in the title or body of data available on the Internet. The problem arises when user is searching for a relevant content yet not sure about the exact terminology to use as a query. The purpose of this study is to help users search for what they mean instead of providing the exact words in the search query. Recently, one of the most popular and new research interests is involving semantic relations between terms or keywords. This method could help users reach the information they are looking for without knowing the exact word or query to search for.

There are several techniques that help the user in searching for information. These techniques are explained in more detail later on in Chapter 2. One method is to expand the search query, which helps the user not only to search in similar contents but also search for broader and narrower contents.

As will be further explained in section 2.1.4, use of thesaurus is one of the common methods for query expansion. Query expansion techniques help user to expand the query with additional keywords in order to narrow down the search results. Unlike dictionary that has definition of terms, thesaurus provides groups of terms according to a degree of relation. Thesaurus is either manually or automatically derived. Automatically derived thesauri are mostly based on statistics and probabilities of terms co-occurrence. Due to calculation costs, automatic thesaurus cannot be applied to very large amounts of documents such as World Wide Web. Addressing this scalability issue is one of the motivations to this study.

1.2 Methodology and Objectives

In this research, the goal is to improve the existing thesaurus-based methods for expanding the search query. These methods are used in order to help users find what they are looking for. There are two main areas that need to be targeted: first the problem of keeping the thesaurus up-to-date, second, considering the terminology relations in different domains.

Manually updating thesaurus is very time consuming and sometimes impractical. There are certain relations between terms that change over time. For instance, the term Obama and President do have a relation now; however, the degree of this relation might not be the same after the next presidential election in the United States.

Since the cost of building manual thesauri is high, in most manual thesauri the relationship between terms are general and the domain is not considered. For example, the term “design” can have different related terms in different domains. In the field of Software Engineering the term “design” may have a strong relation with terms “software” or “algorithm”; while in the field of Interior design, this term can have a strong relation to “pattern” or “decoration”. To solve this problem, a distributed model is proposed that considers term relations in different domains and can be employed in Web 2.0 as well as social networking websites.

In this study we expand the model proposed by De Cock *et al.* [8] in order to build a distributed fuzzy thesauri-based Recommendation System which is scalable. Unlike the proposed model by De Cock *et al.* that tries to analyze all available documents, the proposed method in this study splits the job among the users of social networking sites. Generating the thesaurus for each user based on the local documents reduces the calculation cost for generating and updating the database. The proposed system can be used as a query expansion tool on social networking sites.

It is important to note that there might be other methods such as those implemented by Yahoo! Inc, which may have better results than the method proposed in this study. However, corporations such as Yahoo! that their algorithms are commercialized, do not have all their sources open to public. Thus for the purpose of this study, our work cannot be compared to them. Nevertheless, the work done in this study can be compared to other related academic works published, which will be further explained in section 2.1.4.

1.3 Contributions

The main contributions of this work are as follows:

- Designing and implementing a scalable distributed fuzzy thesauri in order to extract the term-to-term relationships from a number of documents.
- Proposing a distributed Recommendation System that can be used for Web 2.0 and social networking sites, using the above distributed thesaurus.
- Implementing a prototype of the proposed Recommendation System that can be used for expanding search queries.
- Simulating an unstructured P2P network in order to perform scalability test on the proposed distributed model.

Our system will help users to expand their search queries by allowing them to choose from suggested keywords.

1.4 Thesis Outline

The subsequent chapters of this study are organized as follow. Chapter 2 gives an overview of the previous related works on Web2.0, Peer-to-Peer architecture, basic concepts of fuzzy sets and their use in query expansion. Chapter 3 further presents the methodology for a distributed automated Fuzzy thesaurus and query expansion. Next, the implementation, experiments and results are presented and discussed in Chapter 4. The summary of this research and the possible future works are presented in Chapter 5.

Chapter 2

Literature Review

The basic concepts of Fuzzy Sets will be presented in this chapter. The discussion will be then followed by Web2.0 and its applications. Then Unstructured P2P networks and their search techniques will be introduced. In the end, some of the different proposed techniques for Information Retrieval and its applications, also the use of Fuzzy Sets in the Information Retrieval will be reviewed.

2.1 Background Information

2.1.1 Fuzzy Sets

In classical set theory the membership of an elements in a set is determined either by yes or no (1 means yes and 0 means no). In other words, an element is either a member of a set with the degree of one or is not a member with the degree of zero. But the classical set theory barely applies on many of the concepts we deal with everyday. For example, we use terms such as *hot* weather, *high* altitude or *low* pressure, these concepts are considered to be vague and can not be defined in sharp boundaries. In 1965 fuzzy sets were introduced by Lotfi Zadeh [9]. Unlike the classic approach or so called *crisp sets*, instead of yes or no quantification fuzzy sets assess a degree of membership to the elements of sets.

2.1.1.1 Basic Concepts

Fuzzy set A can be defined via its "membership function", which is a set usually denoted by $A(x)$ or $\mu_A(x)$. The membership degree of element x in the fuzzy set A, is defined by the value of $A(x)$. Hence, $A(x) = 1$ indicates x as being a full member of the fuzzy set A, where as $A(x) = 0$ indicates x as not a member of the fuzzy set A. Thus any value between 0 to 1 makes x a partial member of set A.

$$A : X \rightarrow [0, 1] \tag{2.1}$$

The letter X denotes the *universal set* which contains all possible elements of a context or an application that can form a set.

2.1.1.2 Basic Operations

Intersection and Union: In set theory the union of sets A and B is the collection of all distinct elements of two sets and the intersection of two, is the collection of all elements they have in common.

The union and intersection operations in fuzzy sets have the same idea as in the traditional crisp sets. In the fuzzy sets, the same maximum function as in crisp sets is used for Union and the same minimum function is used for Intersection [10].

Given two fuzzy sets A and B the union and the intersection of two sets are defined by the following equations:

Table 2.1: *Union and Intersection of sets A and B in classical set theory*

	x_1	x_2	x_3	x_4
A	1	0	1	1
B	0	0	1	1
$A \cup B$	1	0	1	1
$A \cap B$	0	0	1	1

Table 2.2: *Union and Intersection of sets A and B in fuzzy set theory*

	x_1	x_2	x_3	x_4
A	0	0.15	0	0.85
B	0	0.75	1	0.25
$A \cup B$	0	0.75	1	0.85
$A \cap B$	0	0.15	0	0.25

$$(A \cup B)(x) = max[A(x), B(x)] \tag{2.2}$$

$$(A \cap B)(x) = min[A(x), B(x)] \tag{2.3}$$

Tables 2.1 and 2.2 show an example of these operations in both crisp and fuzzy sets.

2.1.2 Web 2.0

Traditionally Web applications were built to work with read-only services where users mostly access the content of Web sites. But with the advent of social networks, wikis, media sharing sites and blogs the role of Internet users has changed [11]. Unlike the traditional Web, these new services allow users to contribute to the content as well as accessing them. Because of the huge underlying phenomenon there are various definitions for Web 2.0, John Musser in [12] defines it as:

"Web 2.0 is a set of economic, social and technology trends that collectively form the basis for the next generation of the Internet–a more mature, distinctive medium characterized by user participation, openness and network effects."

Although the term Web 2.0 suggests a newer version of World Wide Web, however in reality, this is not the case. Web 2.0 differs from the traditional Web in several ways. Some important differences are [13]:

- Simplifying the Web design concept and making it more flexible by reusing or combining different applications
- Providing a rich user interface
- Facilitation of creation and modification of contents
- Construction of social networks of people that have common interests

There are three main development technologies behind Web 2.0 that developers use in order to deliver rich Web 2.0 applications: AJAX which is the combination of XML and Asynchronous JavaScript, Flex by Adobe and the Google Web Toolkit or (GWT) for short [13].

2.1.2.1 Social Networking Sites

Social networking websites are a good example of Web 2.0 applications. Over the last decade the use of Social Networking sites has dramatically increased among Internet users. A Social Networking site is an online community that allows its users to connect and get in touch with their friends and create groups based on common interests. Some well known examples of such websites are:

- Facebook
- Twitter
- Blogger
- LinkedIn
- MySpace
- Orkut

Based on the type of audience the website is trying to appeal to, they have different functionalities and features. For example MySpace allows more customization and allows its users to design and customize their own HTML pages where Facebook is built to track activities of friends. Other web sites like Youtube and Flickr are only designed to allow members to share media contents such as pictures and videos. The popularity of these websites also varies worldwide, for example Facebook is more popular in North America and Europe but Orkut is popular in South America [14].

2.1.2.2 Small World Characteristic in Online Social Networks

Small World Characteristic is considered an important phenomenon in user network graph of social networking sites. This characteristic was initially explained by Milgram [15] to show how two users or two nodes can be reached on a network by means of the linkage from their neighbors. In order for a network to have Small World Characteristics two conditions must be met. First, the network must have a short path length and second, the clustering coefficient must be high [16]. Hence, each profile page or group page in social networking sites is considered a node.

Consider a blog on blogging social network, called Z, where there are several links to other related blogs. These other blogs are considered the neighbors of Z. Thus the node Z not only has high clustering coefficient, but also it has very small path length. As a result, by clicking on very few number of links, blog Z can be reached, thus it can be concluded that blog Z has small world characteristic. [17].

A research by Mislove *et al.* [18] was performed on LiveJournal - a social networking site - where Mislove calculated the average of short path length as well as clustering coefficient on the network graph. Consequently, an overall average of shorter path length and higher cluster coefficient was observed as compared to web graph. Since privacy issue is an important concern of today's society, there are some social networking sites such as Facebook that do not have the Small World Characteristic.

2.1.3 Peer-to-Peer Networks

In recent years, a major research topic among scientists been the study of Peer-to-Peer architecture, which is considered as a subcategory of Distributed Systems. There are various definitions for Distributed Systems, but generally we can define it as a collection of independent components that appears as one to its users [19]. In traditional centralized approach called Client-Server, a central unit(Server) stores all the information and does all calculations in order to serve its clients. Client-Server approach is very easy to implement that is the main reason that this approach is still widely used. On the other hand this traditional approach has some disadvantages; these systems are more vulnerable to failure due to having one central unit that serves all clients. Another important disadvantage of Client-Server architecture is the server being the bottleneck of the system. Peer-to-Peer networks have several advantages over the traditional approach such as flexibility, efficiency and scalability. Since all peers in a P2P system can act as client and server, they are considered to be inherently scalable and compared to the traditional client and server architectures the possibility of failure or getting compromised by attackers is significantly less. Figure 2.1 compares P2P and Client-Server architectures.

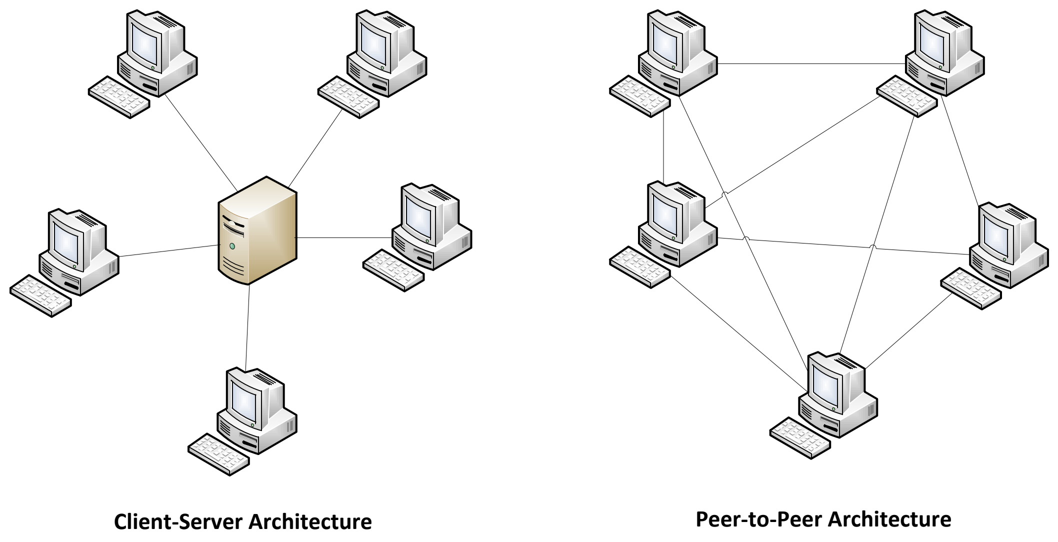


Figure 2.1: *P2P architecture versus traditional Client-Server architecture*

Based on their nature of virtual overlay network, P2P networks are divided into two main categories:

- *Structured:* In this type of P2P systems, a unique subset identifier is assigned to each node which belongs to an identifier space. According to the identifier space, each node has to know about the resources within the corresponding space. Based on this manner, the overlay network is defined in a way that the resources over the network can be found in few hops [20]. Data hash tables or DHTs are one of the most common lookup services being used in such systems. DHTs are similar to hash tables where every node keeps track of only a small number of other nodes on the network by employing a key technique. Chord, Kelips, Tapestry and Kademila are most common DHT protocols being used in today's P2P applications[21].

- *Unstructured:* In pure unstructured P2P systems, there is no particular control over the location of resources and peers do not have track of their neighbors' resources.

Generally, structured systems compared to unstructured ones, are more efficient in finding resources over the network and they produce less *false negatives.* DHT-based lookup services can hit the query in $O(\log(n))$ hops, whereas Gnutella will do it in $O(n)$ hops which can be a huge difference in large scale networks. Although unstructured systems are more capable of resulting *false negatives* but they can handle keyword searches and complex queries better than structured systems. They can also adapt themselves to dynamic environments where peers join and leave the network randomly[22]. Here in this study, the focus will be on several popular searching techniques being used in existing P2P networks and below we provide a brief comparison among them.

2.1.3.1 Search methods in Unstructured P2P Networks

The two major categories of search algorithms in unstructured P2P systems are: *blind search* methods and *informed search* methods. *Blind search* methods do not keep any information about the domain. The search is performed by simply forwarding the search query from one node to another. Generally a parameter called Time-To-Live (or TTL for short) is assigned to each query in order to avoid the huge traffic made by propagating the search query. TTL is a parameter that determines the total number of hops allowed by the system for each query to travel before getting removed from the system. *Informed Search* fashion uses the gathered information from previous queries on the content kept by neighbors in order to utilize the search. Below we will have a brief overview in few algorithm of each category.

2.1.3.2 Blind search methods

Gnutella: This method utilizes a simple *flooding* technique in order to search for contents over the network. Pure flooding technique is simply done by forwarding the search query to all neighbors until the query visits all the nodes on the network. Gnutella works in a same way as the pure *flooding* technique except it assigns a TTL parameter to each search query which decreases by one in every hop it travels. The search continues until the value of TTL is decreased to 0; at this point, system discards the search query. Using TTL improves the amount of traffic compare to the pure flooding. Although this method is simple and sounds efficient, it is found that it creates a huge traffic when the network gets larger. So this matter makes it impractical when the network scales[22].

Modified-BFS[23]: This method is very similar to the traditional flooding technique, except it forwards the queries to the random number of neighbors instead of all. Compared to the traditional Gnutella it produces less traffic but still carries the same problem that Gnutella does which makes it not scalable.

*Random Walks [24]:*In this method, the query is sent to k random number of neighbors. Each query is called a *walker* and travels its own path. The *walker* dies either by using TTL technique or by *checking* method which query contacts the query sender to see if the termination condition is met or not. This method in comparison to other methods produces less traffic in the network ($k * TTL$ in the worst case using TTL) but the main disadvantage of using this method is its variable performance which is because of its random nature . According to different conditions the performance of this method might be different. Another disadvantage is that the queries of popular contents and rare object are treated in the same way since there is no learning in this method [25].

2.1.3.3 Informed Search Methods

Super-Peer Approaches: In *super-peer approach* there are super-peers acting as proxies for other peers which are connected to them as leaves. As one leaf sends a query to its super-peer, the super-peer sends the query to the neighboring super-peers as well as its own related leaves. This process is similarly repeated by other super-peers. Since the queries are sent only to relevant peers, as a result, no extra nodes will be visited. These super-peers exchange information and indexes regularly in order to prevent overhead in the network and avoid unnecessary forwards as much as possible. These networks are similar to hybrid networks in terms of reducing the effect of flooding by utilizing it. In order to increase the number of query hits it is very vital that regardless of how dynamic the network is, the number of leaves connected to super-peers has to stay online all the time[26].

Intelligent-BFS:[27] This method is the intelligent version of blind BFS, except it keeps track of previous searches made or passed from each node. Each node tries to rank its neighbors based on the previous experiences which can be used to determine the possibility of getting a query hit for each query. The main disadvantage of using this method is the overhead created by nodes updating their index table in order to keep the track of their neighbors.

Distributed Resource Location Protocol[28]: In this method, the location of objects on the network are unknown to the peers. Nodes send search query to their neighboring nodes according to a particular probability. If the query is hit, the query takes the same path to go back to the initiator peer thus the nodes on its way back will store the location of that object. Thus in the case where those peers receives the same query, they will refer to the peer that found the object. If the requester peer already obtained the object then the second search is successful in one step, otherwise another blind search will be initiated.

2.1.3.4 Comparison Between Search Methods

D. Tsoumakos and N. Roussopoulos in [26] have compared above search methods in terms of “success”, “bandwidth-efficiency” and “adaptation to dynamic network conditions” where nodes join and leave the network frequently. In cases where at least one object is found, the search is considered successful. A query may have multiple discoveries, the number of discovered objects is called *hits*. From this comparison they concluded that in most cases simple blind search methods are more suitable. Although informed search methods have shown to be more accurate in terms of accuracy but because of their complexity and the cost of maintenance they found simple blind search methods more suitable. They are also shown to be relatively stable in both static and dynamic environment.

2.1.4 Query Expansion in Information Retrieval Systems

For each term in a language there might be more than one other term with similar meaning available. Suppose there exist two terms t_1 and t_2 with similar meaning in one context yet different in another. Imagine only the term t_2 appears in a document d. Once a user searches for term t_1 among all documents, the system would not include document d in the search results; even though the document d contains the term t_2 that is in the same context as t_1. In Information Retrieval, this issue is known as “Synonymy” [29]. A good example of this situation could be the similar meaning between the terms “aircraft” and “plane”. Although the term *plane* in the same context refers to *airplane*, yet in a different context it could refer to a flat surface. This issue can be addressed by user refining the search query where the system can be used to provide recommendations to the user in the following two categories: local or global which are briefly explained as follows.

2.1.4.1 Local Approaches

In this method, the search starts by user entering an initial query. Based on the matched documents with higher ranking, terms are added from the relevant documents in order to expand the search query [29]. There are three main approaches for the Local method:

1. **Relevance Feedback:** This method requires involvement of the user in the process of information retrieval. User is asked to provide proper feedback on how relevant or irrelevant the returned results are. Often, the user may not be familiar with the collection, thus iteration of the refinement process may be required to reach the best results. The relevance feedback approach is done through a series of steps.

    - First the user enters an initial query.
    - System then returns initial sets of results.
    - Next the user ranks the returned results as either relevant or irrelevant.
    - According to Walker [30] the system “re-weight, expand or reformulate” a new search query based on the feedback received from the user.
    - The new improved representation of the results are shown.

- If the results are still not satisfactory, the Relevance Feedback process can be repeated.

There are some factors that affect the quality of results returned by Relevance Feedback method.

(a) User must have a general idea about the information he or she is seeking in order to be able to properly mark the relevant or irrelevant items from the search results. Relevance Feedback without the help of other methods has some disadvantages.

  - Since the system searches for the exact terms entered by the user, in case of spelling mistakes the effectiveness of this method can be questionable.
  - As documents in different languages are placed in different clusters, searching for a term in a language does not necessarily return related documents in other languages.
  - Using different vocabularies for a term can make the searching process a bit tricky. For instance, the two terms *airplane* and *aeroplane* both have the same meaning. Yet if the user searches for the term *aeroplane* in a document that only has the term *airplane*, then no results would be returned.

(b) Clustering documents based on their similarities is needed in order to provide relevance feedback. This approach performs well if the similarity of the relevant documents to non relevant documents chosen by the user is low or in another word, their clusters do not have much overlap.

2. **Blind Relevance Feedback:** This approach is also known as pseudo-relevance

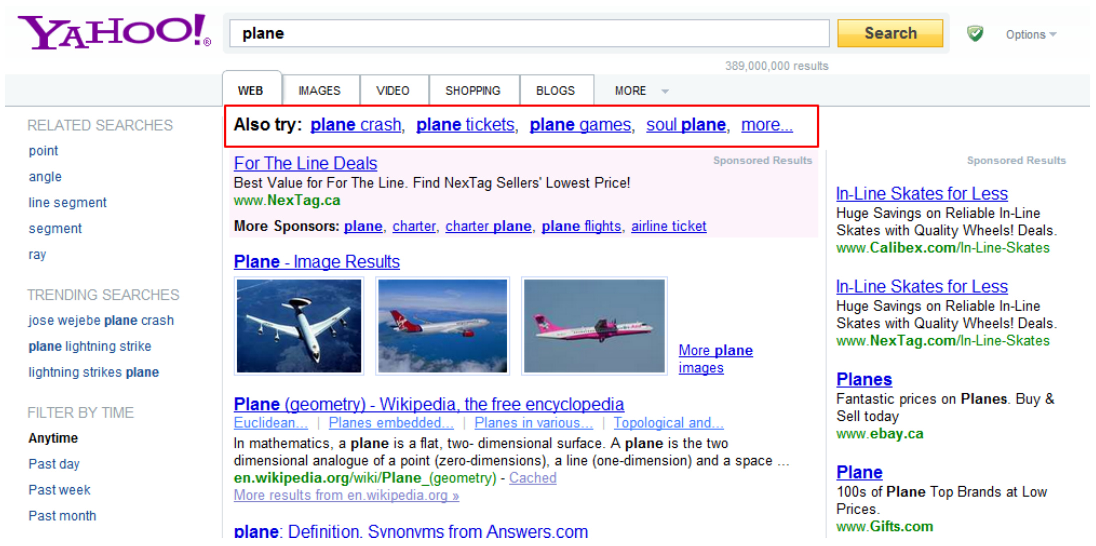
feedback. Similar to Relevance Feedback approach, this method performs the initial retrieval. Yet in the next step instead of asking the user to choose the relevancy of returned results, blind relevance feedback method takes the top n documents that have been ranked as relevant. Thus by making the algorithm completely automated, it eliminates the user involvement in query refinement. However, according to Walker [30] the main drawback of blind relevance feedback is the case where the selected top n documents are not relevant. Consequently, the system would make expansion based on non relevant new terms; thus the effectiveness of this approach becomes questionable.

3. **Indirect Relevance Feedback:** This method is also known as implicit relevance feedback [29]. In this approach, previously collected data on statistics such as number of click on links are used to evaluate the relevancy of documents in search results. Although this approach is not as accurate as relevance feedback method introduced earlier, yet it is more reliable than the blind relevance feedback approach; which does not consider the user's opinion.

2.1.4.2 Global Approaches

Unlike the local methods introduced, Global approaches do not depend on the search query itself or its results. Global approaches include query expansion and other techniques such as spelling correction. The spelling correction techniques refine the search query by helping the user to correct typo or misspelling errors. Whereas query expansion refines the query by expanding the search keywords and adding extra terms. This is unlike relevance feedback approach of Local methods, where change in the weight of terms in search query adjusts the returned results.

Search engine websites are great examples of where query expansion method is applied. For instance, Figure 2.2 shows the recommendations by Yahoo! search engine [31] for the term *plane*. The extended query recommendations as shown are *plane crash*, *plane tickets*, *plane games* and *soul plane*.



plane

389,000,000 results

RELATED SEARCHES
point
angle
line segment
segment
ray

TRENDING SEARCHES
jose wejebe **plane** crash
plane lightning strike
lightning strikes **plane**

FILTER BY TIME
Anytime
Past day
Past week
Past month

Also try: **plane** crash, **plane** tickets, **plane** games, soul **plane**, more...

Sponsored Results

For The Line Deals
Best Value for For The Line. Find NexTag Sellers' Lowest Price!
www.**NexTag.ca**

More Sponsors: **plane**, charter, charter **plane**, **plane** flights, airline ticket

Plane - Image Results

More **plane** images

Plane (geometry) - Wikipedia, the free encyclopedia
Euclidean... | Planes embedded... | Planes in various... | Topological and...
In mathematics, a **plane** is a flat, two- dimensional surface. A **plane** is the two dimensional analogue of a point (zero-dimensions), a line (one-dimension) and a space ...
en.wikipedia.org/wiki/**Plane**_(geometry) - Cached
More results from en.wikipedia.org »

plane: Definition. Synonyms from Answers.com

Sponsored Results

In-Line Skates for Less
Huge Savings on Reliable In-Line Skates with Quality Wheels! Deals.
www.**Calibex.com**/In-Line-Skates

In-Line Skates for Less
Huge Savings on Reliable In-Line Skates with Quality Wheels! Deals.
www.**NexTag.com**/In-Line-Skates

Planes
Fantastic prices on **Planes**. Buy & Sell today
www.**ebay.ca**

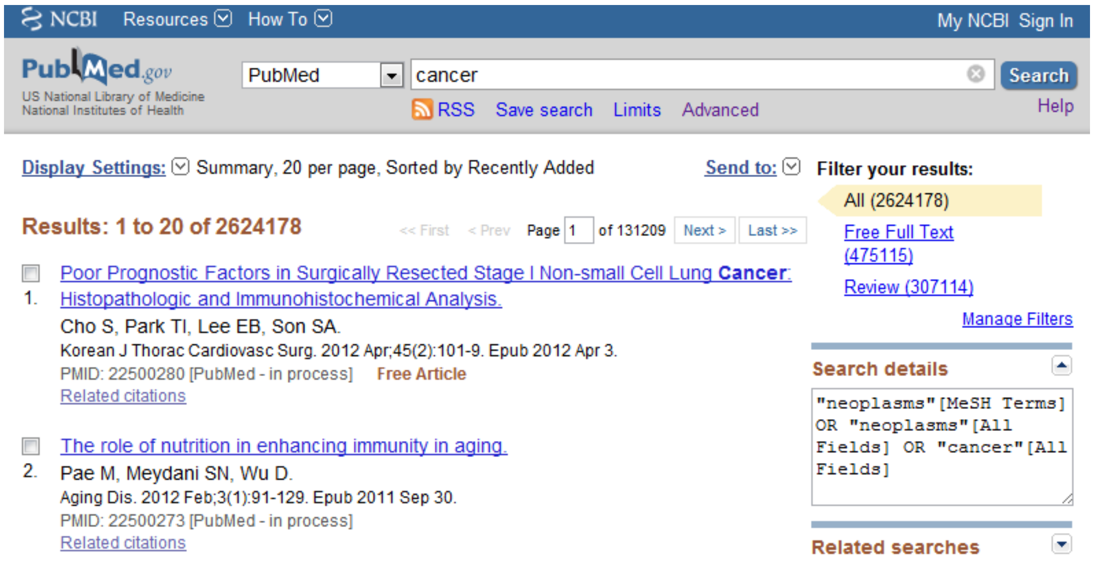
Plane
100s of **Plane** Top Brands at Low Prices.
www.**Gifts.com**

Figure 2.2: *Query expansion recommendations by Yahoo! search engine website in 2012*

In order to generate recommendations to expand the query, tools such as thesaurus are utilized. The general idea is using synonyms and words similar in meaning to the term entered by user in order to provide more efficient and relevant search results.

1. **Controlled Vocabulary:** In specialized field such as medicine where enough information about the context of the field is available, the method of controlled vocabulary is applied. In this approach, terms that belong to the same concept

are grouped together under the same heading by human experts. PubMed is an example of search engine used to explore medical literatures in MedLine database. As shown in Figure 2.3 PubMed website utilizes "Unified Medical Language System (UMLS)" [32], which provides a controlled vocabulary database in the field of biomedical science. Thus if a user searches for the term cancer, the system automatically adds the term "neoplasms" referring to abnormal growth of tissue, which is a characteristic of cancer to the search query. This step is performed without the user's involvement and the user might not even realize that this step was performed. Thus a human expert must manually create a database where the terms cancer and neoplasms are related in the concept of medicine.



Display Settings: Summary, 20 per page, Sorted by Recently Added

Send to:

Results: 1 to 20 of 2624178

<< First < Prev Page 1 of 131209 Next > Last >>

1. Poor Prognostic Factors in Surgically Resected Stage I Non-small Cell Lung **Cancer**: Histopathologic and Immunohistochemical Analysis.
Cho S, Park TI, Lee EB, Son SA.
Korean J Thorac Cardiovasc Surg. 2012 Apr;45(2):101-9. Epub 2012 Apr 3.
PMID: 22500280 [PubMed - in process] Free Article
Related citations

2. The role of nutrition in enhancing immunity in aging.
Pae M, Meydani SN, Wu D.
Aging Dis. 2012 Feb;3(1):91-129. Epub 2011 Sep 30.
PMID: 22500273 [PubMed - in process]
Related citations

Filter your results:
All (2624178)
Free Full Text (475115)
Review (307114)
Manage Filters

Search details

```
"neoplasms"[MeSH Terms]
OR "neoplasms"[All
Fields] OR "cancer"[All
Fields]
```

Related searches

Figure 2.3: *PubMed website expands the search query using UMLS controlled vocabulary database*

2. **Query Expansion Using Thesaurus:** There are two types of thesauri, handcrafted or manual thesauri and automatic thesauri.

   (a) *Manual thesaurus:* Manual thesauri also known as handcrafted thesauri are built by human experts and contain hierarchies of associated terms. Although manually created thesaurus is more accurate in terms of quality of term relations as compared to automated thesaurus; however, there exist some disadvantages to this approach.

   As Schutze and Pedersen [33] mentioned, a thesaurus should be broad enough to cover all data structures and ensure the "semantic relatedness between words". Yet, thesauri that are not specific to a field or domain such as Rogets, as Walker states in [30] are "thought to be too broad to be useful" in systems such as text retrieval. Thus according to McCune in [34], thesauri that are based on a particular domain are found to be more useful.

   The initial creation of domain specific thesaurus is quite costly as it takes a lot of time and effort for human experts to build. On the other hand, in the fast evolving disciplines such as science, the maintenance cost of keeping manual thesaurus up-to-date is quite high. These disadvantages of manual thesauri have encouraged computer scientists to investigate alternative automatic solutions.

   (b) *Automatic thesaurus:*Automatic thesaurus on the other hand is a collection of related terms that as Walker mentions in [30], are "derived from statistical word co-occurrence or lexical relationships". Generally the process of generating automatic thesaurus has three phases:

   - A vector space model is created by extracting the term co-occurrences

- Similarities between terms are calculated based on the vector space model
- Apply a clustering technique based on term-to-term relationship

There are several approaches in measuring terms co-occurrences. For instance in studies done by Pereira *et al.* [35], Ruge [36] and Lin [37] the term co-occurrence is calculated based on the neighboring terms. Whereas in study by Sanderson and Croft [38], co-occurrence is calculated based on the whole document.

2.1.5 Indexing

One of the most common ways to speed up the search process is building a structure from the data that is referred as indexes. Dealing with a large data structures with semi-static properties indicates the benefits of building and maintaining an index related to that data. Semi-static is refereed to collections that do not change frequently. It should be noted that usually collections with semi-static properties need be updated regularly within a reasonable interval. There are lots of different indexing approaches that are applied depend on problems and situations [39].

2.1.6 Clustering

Clustering enhances the process of information retrieval by grouping similar objects together. The clustering techniques used with goal of making improvement in efficiency and effectiveness of Information Retrieval procedures.

With this point in mind that class identification is not required before processing them, clustering can be considered as a procedure that develop some structures in the large data sets. Moreover, because there is a chance that clustering reveals some hidden relationships between data, it is considered as a discovery tool among some scientists[40].

2.2 Related Works

2.2.1 Fuzzy Clustering Methods

In information retrieval system, calculating the degrees of relevancy among documents is necessary. Thus the main goal is retrieving documents that have higher degree of relevancy to user search queries. The performance of good retrieval system is highly restricted to the ability of describing user request by a query. Nowadays, the keyword-based text search methods used in most commercial Information Retrieval Systems is the boolean logic model. Yet, the biggest problem with Boolean Logic in the systems is the fact that user queries cannot accurately be characterized by the index terms. Generally terms that are entered by users in search fields have some kind of fuzziness in them that must be addressed adequately. It is important to know the origin of fuzziness in user's queries. There is a great chance that the user is unfamiliar with the subject he or she is trying to search for. Also the user may have the same unfamiliarity toward the correct use of information retrieval system.

So, in general Fuzzy Clustering method are performing better than a traditional crisp clustering algorithm, especially when the boundaries between clusters are not crisp or in other word documents clusters are overlapping. More over Fuzzy clustering algorithm by handling overlapping cluster present more flexibility that a traditional crisp clustering algorithm[41].

2.2.2 Information Retrieval in Web2.0

Social Media is one of the most popular online platforms that shaped by information sharing concept between users of this type of online platform. MySpace, Facebook and

Twitter are well know examples of this type of platform. Information Retrieval from this type of platform has some advantages. Mainly, the importance of keywords based on the popularity of a person as well as the similarity of keywords among users is easy to recognize in Information Retrieval systems that are based on social media. Moreover, the fact that social media based Information Retrieval Systems can give higher priority to users in social networks is an advantage of these systems. It should be noted that these advantages could be considerable during the comparison of traditional Information Retrieval Systems with Social Media Information Retrieval Systems. Lee *et al.* in [42] have proposed a new Information Retrieval model that involves social networking trends as well as user's context in order to improve the traditional ranking methods. The results of their study shows that context-awareness Information Retrieval is very promising in social networks. Context-awareness is simply referred to the idea of characterizing the situation and making reactions based on the situation. However, their proposed algorithm does not provide a good performance dealing with mass data sets.

Recently researchers have tried to extract information based on friendship networks and user interests from Facebook and Twitter. He *et al.* in [43] utilize the collected information as the basis for recommendations to the users who have the same interests or friendship networks. Based on real data obtained from social networking sites, their study shows that immediate friends in social networking sites tend to provide similar feedbacks and reviews on different matters. The study also shows that distance friends (friends of friends) have the same characteristic as immediate friends but with the lower degree. In their research they have considered immediate friends as well as distance friends to form a probabilistic model in order to provide recommendations that the user might be interested in. He and his colleagues have claimed that their proposed recommendation system can provide relatively accurate predictions and the performance is reasonably practicable. However the problem arises due to the fact that such initiatives are based on probabilistic models and assumption that data available are accurate. However, the data that users add to social networking sites are perception-based and can be imprecise. To clarify the problem we can consider the following example: if a user says, "I usually drink Tim Horton's coffee" or "I like Toronto", this data cannot be used in a simple frequency-based probabilistic model; since the meanings of such term as "usually" or "like" vary among users. In early and even more recent approaches such relations between terms are assumed to be made by human experts. The problem with manual approaches is that they can hardly be called flexible since they take a significant amount of time and effort to build; thus can only be applied to a limited number of documents.

2.2.3 Use of Fuzzy Sets in Information Retrieval

The role of Fuzzy Set Theory in information retrieval is crucial. According to Pasi [44], Fuzzy Set Theory has been applied in several cases as following:

- Utilized to characterize new Information Retrieval Systems
- Employed to solve the subjectivity and impreciseness of information in document indexing
- To control the "vagueness" of search terms entered by the user
- To design a more flexible system, for example in thesauri algorithm used to extend information retrieval search engines

One common method of clustering is through the provided citations in document pairs. These clusters are utilized as indexing for information retrieval. On that note, through the application of fuzzy clustering, the association to each cluster is not limited to a single document, hence, multiple associations can exist for a single document [45]. In general, the pure fuzzy clustering technique allows for a full overlap of clusters where the modified fuzzy clustering provides threshold mechanisms to limit this phenomenon.

Nevertheless, Fuzzy Set Theory plays an important role in the field of Information Retrieval as it defines flexible systems by means of managing subjectivity and vagueness of information [45]. Adding fuzzy-logic-based methods to search engines has already been proposed by several researchers. The suggestion was introduced ever since the existence of imprecise and perception-based data on the classical web was identified. For instance, Lotfi Zadeh in [46] suggested a fuzzy approach to upgrade the existing search engines into question-answering systems that have the deduction capability.

In another work, which is based on Zadeh's question-answering model, Korotkikh [47] has proposed a new fuzzy spectral approach for search engines in order to integrate information in search engines. De Cock and his colleagues in several works (the latest is [8] which will be explained in Chapter 3) proposed building a thesauri fuzzy system that can be employed on top of current search engines by adding an extra layer of term-to-term relationship which can be extracted from term-to-document relationships.

Miyamoto in [48] developed a fuzzy based model on information retrieval based on obtaining three distinct components: a fuzzy association, a fuzzy inverted index and a fuzzy filter. In this study, Miyamoto developed a large scale bibliographic database through development of an efficient algorithm based on associations. The fuzzy set model proposed by Miyamoto provides a "clearer view" compared to the traditional approaches that use weight. However Miyamoto's model had some limitations that required further studies; problems such as applying the model in related areas such as "structure of texts" and "bibliography" or providing efficient and scalable algorithms that can be used in large databases.

Nomoto *et al.* in [49] used fuzzy based model and worked toward the formation of a fuzzy retrieval system that is solely based on citations. In this case, the citations are classified in a Boolean formation; however, their distinctions are made via the application of fuzzy graph theory. According to [49], their proposed method is scalable and practical to be used in the fields that involve inheritance.

Beigbeder *et al.* in [50] have presented a new mathematical model for Information Retrieval. In the model proposed by Beigbeder and his colleagues they have used fuzzy sets in order to expand the traditional boolean model used for document ranking. Using the expanded boolean model allows them to consider the position of the search keywords while calculating the relevance degree of documents and the search keywords. For example if the user is searching for "apple computer" and the system returns two documents containing both keywords, the system will assign a higher rank to the one that the position of these two keywords are closer. Since this method does not require the entire collection of documents in order to generate the values, it can be used in larger scales with documents distributed over a network.

As previously discussed in this chapter, using controlled vocabulary can be an effective tool for improving the search results in specialized fields such as medicine. Controlled vocabularies can also be seen as ontologies in a sense that the terms grouped together represent the same concept in a particular domain. Parry in [51] proposed a new model that extends the use of controlled vocabulary systems such as UMLS [32] that is already discussed in this chapter. According to Parry, the importance of mappings for terms in a controlled vocabulary can vary for different users of the system. In order to overcome this problem, relevance feedback methods were used in order to create a "fuzzy ontology" which a membership degree is assigned to each term in each group. In order to test the usability of the proposed method, Parry has implemented the system and asked a small group of users to use it for a while. According to Parry, the generated database can potentially be used in search engines.

Chapter 3

Model and Methodology

In this chapter, the proposed methodology for distributing the Fuzzy Thesauri's knowledge-base over unstructured P2P networks will be presented. The unstructured P2P network can be obtained from the friendship graph of social networking sites. In Chapter 4 the scalability of the proposed model will be tested and the results will be presented.

3.1 Introduction

This study uses a similar method as to De Cock[8], which will be explained in more detail. The term-to-document relationship can be built by common search engines, and usually creates a weight matrix using frequency-based probabilistic method. This algorithm is quite simple; it calculates term frequency in a document by dividing the number of repetitions of that term by the total number of the words. By using the term-document weight matrix, a fuzzy approach method based on [8] is applied in order to extract the term-to-term relationships, which shows how two terms are related to each other. In other words, based on the available documents and their contents, the degree to which a word is related to other words are measured. For example, the word "apple" maybe related to "company" with the degree of 0.7 and maybe related to the word "computer" with the degree 0.3, based on the content of the documents that are analyzed. Therefore according to the generated knowledge-base on the analyzed documents, when a user enters the word "apple" the Recommendation System suggests the word "company" first because it has a higher rank compared to "computer".

As previously explained in Chapter 2, this type of hint can be very useful for domain-specific documents but not the entire Web. There are reasons why building a fuzzy search engine for the entire Web is impractical. Considering the term-to-term matrix calculation that will be discussed in this chapter, it is clear that for each new document added to the system, the term-to-document matrix and therefor the term-to-term relations require recalculation. In fact, due to the large amount of documents on the Web, none of the proposed methods for the fuzzy thesauri could be applied for the Web.

3.2 Proposed Model

A term-to-term relationship matrix was built from regular words based on the available documents on social networking and Web2.0 applications. The employed method is similar to what has been previously proposed by De Cock *et al.* [8]. However as previously explained, due to the scalability issue it is impractical to employ this method for the entire Web. In this study we propose a distributed solution that builds the knowledge-base separately for each node on a P2P network. As discussed in Chapter 2 it is assumed that mostly the users of social networking sites have common interests. Therefore, the knowledge-base generated based on local documents is expected to be domain specific. However, there might be cases that two users do not have common interests in a particular field. We also propose that the social networking sites employ a P2P network in order to store the knowledge-bases in each peer in a distributed manner.

The user sends the query to a P2P network consisting of servers (i.e. peers); each of these servers has its own knowledge-base, which was built using the documents available on that specific server. For example, the knowledge-base for the computer engineers' network in LinkedIn has different term-to-term relation in comparison to that of the LinkedIn network for computer scientists. Whenever a user who is searching for a job in computer engineering types the word "design," only the related words such as "chip design" or "IC design" should be shown by the Recommendation System, whereas when a job seeker who belongs to the computer scientists' network types "design," the suggested related words might be "algorithm design" or "software design," etc. It must be noted that in order to respect users' privacy, only publicly available information will be used for building the knowledge-bases.

The system takes the following steps in order to provide recommendations for a term t:

1. In each node when the user types in a keyword t in the search field, the system sends the query to all nodes which are directly connected to it.

2. When a node receives a query, it searches in its knowledge-base and sends the results back to the requesting node in the form of three separate sets plus the number of documents that its knowledge-base is generated based on. Bean and Green in [52] have explained the different types of the relationships that terms may have. The three forms of such relationships used in this study are listed below:

    - Related Terms (RT): The list of words that are generally related to t. This relation is neither "synonym" nor "hierarchical".

- Broader Terms (BT): List of terms that their scope of meaning includes the scope of term t.
- Narrower Terms (NT): List of terms that the scope of the term t includes their scope of meaning.

The following examples show the Broader and Narrower Term relationships. Consider the three terms: "Canada", "Ontario" and "Toronto". "Canada" includes "Ontario" means "Canada" is a Broader Term for "Ontario". Similarly "Ontario" includes "Toronto" means "Ontario" is a Broader Term for "Toronto". Now consider the Narrower Terms: in this case "Toronto" is a Narrower Term for "Ontario" and "Ontario" is a Narrower Term for "Canada".

3. When the requester node receives the recommendation results from other nodes, it merges all lists and creates a list of unique terms. Then it normalizes the relationship degree for each term based on the number of its appearance among all lists returned by other nodes and its relationship degree within each list.

3.2.1 Employing Fuzzy Thesauri in Recommendation System on Each Node

Phase 1: calculating the fuzzy document-term relation: The term-document weight matrix specifies the relationship between terms and documents. The term-document relationship matrix is a $n * m$ matrix where n is denotes the total number of terms and m denotes the total number of documents and can be calculated using any probabilistic manner. In this study we use $tf - idf$ [53] which is one of the most common methods to calculate the term-document weights. This method also suggested by [8] because of its simplicity. The algorithm 3.1 is quite simple and can be calculated using the following equations:

$$tf - idf(t, d, D) = tf(t, d) \times idf(t, D) \tag{3.1}$$

Where tf or term frequency is the number of times that the term t appeared in document d divided by the total number of words in the document d and D denotes the set of all documents.

The inverse document frequency or idf is a measure of the importance of the term t among the set of all documents D, which can be calculated using the following formula:

$$idf(t, D) = \log \frac{|D|}{|d \in D : t \in d|} \tag{3.2}$$

It is important to note that since tf_idf is based on statistics, it might not produce accurate results when it is applied on very short texts. Therefore, our proposed system may not be a good option for social networking sites such as Twitter that have limitation for the length of documents.

Since $tf - idf$ is calculated based on the total number of documents, therefore it has to be regenerated each time a new document is added to the system in each node. One task of the proposed Recommendation System is to do such update for each node.

Algorithm 3.1 displays the steps in generating the term-document matrix. All terms in available documents are extracted and added to the dictionary one by one. In the case where the term already exists in the dictionary, the count number is increased. Also in a separate table the terms and the document numbers are stored. As soon as the complete list is collected, the value of $tf - idf$ for each term in each document is calculated and will be stored in the database. The notations used in the algorithms are presented in

Table 3.1: *Notations used in the algorithms.*

Notation :	Definition
documents :	Set of all local available documents
ExtractWords() :	This method takes a document and returns the extracted words as an array.
AddToDictionary():	This method takes a word and adds it to the dictionary
SetWordCount() :	This method takes a word and its count showing how many times it appeared in all documents.
AddToTermDocumentTable() :	This method takes a word and a document id and assigns them together
GetWordsInDocument() :	This method takes a document and returns the set of all unique words which the document contains
CalculateTF() :	This method calculates the frequency of the term t in the document d
CalculateIDF() :	This method calculates the idf value for the term t in the collection of all documents D
UpdateTermDocumentTable() :	This method takes document_id, word and the tf_idf and updates the row.
FuzzyCardinality() :	Takes a fuzzy set as an input and calculates the cardinality measure.
FuzzyIntersection() :	Calculates the fuzzy intersection for the sets A and B.
AddToThesaurus() :	Adds the entered words to the thesaurus knowledgebase.

table 3.1.

Phase 2 calculating the fuzzy Similarity and Inclusion measures: Since we have the document-term relation matrix we can transform terms into fuzzy sets. For every term t we can define a fuzzy set W_t as the set of documents relevant or related to term t. Now finding the degree of association between two terms is a matter of the relatedness between their fuzzy sets [8].

Similarity and inclusion measures have been two important topics after Zadeh introduced fuzzy sets. The inclusion measure refers to the degree that how much one fuzzy

Algorithm 3.1 Generating the Term-Document Matrix

```
Require: documents
 1: for all document in documents do
 2:    // extract all terms from the document
 3:    words = ExtractWords(document)
 4:    for all word in words do
 5:       if word does not exist in the dictionary then
 6:          AddToDictionary(word)
 7:       else
 8:          current_count = GetExistingCount(word)
 9:          SetWordCount(word,current_count + 1)
10:       end if
11:       AddToTermDocumentTable(term,document)
12:    end for
13: end for
14: for all document in documents do
15:    words_in_document = GetWordsInDocument(document)
16:    for all word in words do
17:       // calculating the term tf_idf and inserting the value along with the term into
          the database
18:       D = total number of documents
19:       tf = CalculateTF(term,document)
20:       idf = CalculateIDF(term,document,D)
21:       tf_idf = tf * idf
22:       UpdateTermDocumentTable(document,term,tf_idf)
23:    end for
24: end for
```

set is contained in another one. The similarity measure simply refers to the degree of similarity between two fuzzy sets [54]. The fuzzy Similarity and Inclusion measures can be applied on the fuzzy sets A and B as follows:

$$Sim(A_{t_1}, B_{t_2}) = \frac{\sum_{i=1}^{n}(A_{t_1}(x_i) \wedge B_{t_2}(x_i))}{\sum_{i=1}^{n}(A_{t_1}(x_i) \vee B_{t_2}(x_i))} \tag{3.3}$$

$$Inc(A_{t_1}, B_{t_2}) = \frac{\sum_{i=1}^{n} (A_{t_1}(x_i) \wedge B_{t_2}(x_i))}{\sum_{i=1}^{n} A_{t_1}(x_i)} \tag{3.4}$$

Where A and B are fuzzy sets for the terms t_1 and t_2 and the two operands $\wedge$ and $\vee$ express the minimum and the maximum respectively. It is also assumed that A and B are not empty meaning that each term is related to at least one document to the degree greater than zero.

Algorithm 3.2 Generating the Term-Term Relationships

```
Require: words
 1: for all word1 in words do
 2:    for all word2 in words do
 3:       // creating the fuzzy sets of related document for word1 and word2
 4:       A = GetRelatedDocumentsFuzzySet(word1)
 5:       B = GetRelatedDocumentsFuzzySet(word2)
 6:       // calculating the fuzzy similarity and inclusion
 7:       similarity = FuzzyCardinality(FuzzyIntersection(A,B)) / FuzzyCardinality(FuzzyUnion(A,B))
 8:       inclusion = FuzzyCardinality(FuzzyIntersection(A,B)) / FuzzyCardinality(A)
 9:       saving the results into the thesaurus knowledge-base
10:       AddToThesaurus(word1,word2,similarity,inclusion)
11:    end for
12: end for
```

Algorithm 3.2 displays the steps in generating the term to term relationship. Now that the database is available by employing algorithm 3.1, all possible pair of words are generated though a nested loop. For each pair the degree of similarity and inclusion are calculated and stored in thesaurus. The thesaurus forms a $n*n$ matrix(where n indicates the number of terms) that will be stored in a relational-database.

For example considering a set of documents D consist of 8 documents $D = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\}$ and two fuzzy sets A_{t_1} and B_{t_2} as shown in the Table 3.2. Where A_{t_1} is

Table 3.2: *An example of fuzzy sets for the terms t_1 and t_2*

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
A_{t_1}	0.4	0	0.6	0.1	0	0	0.75	0
B_{t_2}	0.55	0.45	0	0	0	0	03	0.1

a set containing all documents related to the term t_1 and B_{t_2} is a fuzzy set containing all documents related to the term t_2. Using the formulas 3.3 and 3.4 the similarity between the two fuzzy sets can be calculated as follows:

$$Sim(A_{t_1}, B_{t_2}) = \frac{(0.4 + 0.3)}{(0.55 + 0.45 + 0.6 + 0.1 + 0.75 + 0.1)} = 0.27 \tag{3.5}$$

This value determines the degree of relationship between the two fuzzy sets A and B that is interpreted as the similarity between the two terms t_1 and t_2.

Unlike the *Similarity* measure, the *Inclusion* measure is not symmetric, so the value of $Sim(A, B)$ can be different than the value of $Sim(B, A)$. The inclusion measure for the two fuzzy sets can be calculated as follows:

$$Inc(A_{t_1}, B_{t_2}) = \frac{0.4 + 0.3}{(0.4 + 0.6 + 0.1 + 0.75)} = 0.37 \tag{3.6}$$

$$Inc(B_{t_2}, A_{t_1}) = \frac{0.4 + 0.3}{(0.55 + 0.45 + 03 + 01)} = 0.5 \tag{3.7}$$

Since the value of $Inc(B_{t_2}, A_{t_1})$ is greater than the value of $Inc(A_{t_1}, B_{t_2})$, in can be concluded that the fuzzy set B_{t_2} includes the fuzzy set A_{t_1}. Therefore, we can interpret that as the term t_2 includes the term t_1 with the degree of 0.5.

The normalization for the term k from the merged result set can be calculated using the following formula:

$$Norm_k = \frac{n}{N} * \sum_{i=1}^{n} (\frac{R_i * P_i}{R}) \tag{3.8}$$

In formula 3.8, k denotes the selected term from the merged list, N indicates the total number of lists received, n is the number of lists contain the term t, R_i is the number of documents which the set i is generated base on, R is the total number of documents reported by all nodes and finally P_i is the relationship degree between the searched term t and the term k in the set i.

Consider an example where a user from a node called "Local" sends a search query for the term "design" to its neighboring nodes by typing it in the search field. Consequently, three neighboring nodes sent back results to the initiator node. Table 3.3 illustrates this affiliation. The values in the parenthesis show the relationship degree between each term and the searched term. For instance, the term "software" in the first row of the node "Local" has 0.6 degree of relation to the term "design". Finally, the four lists under the nodes "Local", "Neighbor 1", "Neighbor 2" and "Neighbor 3" are combined together to generate the final list that will be displayed to the user. In order to generate the final list, equation 3.8 is used. This final list is shown in Table 3.4.

Table 3.3: *An example of distributed search results*

	Local	Neighbor 1	Neighbor 2	Neighbor 3
1	software(0.6)	software(0.7)	station(0.85)	algorithm(0.75)
2	pattern(0.55)	algorithm(0.6)	city(0.65)	software(0.7)
3	algorithm(0.4)	pattern(0.55)	software(0.4)	plan(0.65)
4	network(0.35)	map(0.35)	car(0.25)	network(0.5)
5	circuit(0.2)	plan(0.2)	school(0.2)	map(0.3)
No. Docs	20	35	10	30

Table 3.4: *An example of normalized results*

Rank	term	Overall Score
1	software	0.64
2	algorithm	0.45
3	pattern	0.29
4	network	0.22
5	station	0.21
6	plan	0.203
7	map	0.175
8	city	0.16
9	car	0.06
10	circuit	0.05
11	school	0.05

Now user can select the term “software” in order to expand the query as “design software". After the user adds a new word to the query the Recommendation System will be triggered again and provides new recommendations to the user. Let's assume that the user selects the term "ontario" from the recommendation list. In the next run the Recommendation System shows a sorted list of related terms for "'ontario' where the term "'toronto' is on the top. Therefore the user can expand the query as "design software ontario toronto" and submit it in order to retrieve the related documents.

3.3 Underlying Network

As the Recommendation System cannot be used by regular search engine methods such as indexing due to scalability issue, the remaining distributed methods that might be suitable for the proposed system are DNS [19] and P2P approaches. Since the purpose of this study is to generate a domain specific thesaurus, it is required to take advantage of social network characteristic, which is individuals with the common interests being linked to one another. Therefore, in this study the graph of social networking sites is considered as an unstructured graph and unstructured P2P network searching methods were applied.

3.4 Summary

In this chapter, an automated fuzzy thesauri based Recommendation System were introduced. The proposed method will use the friendship network of social networking sites in order to suggest domain specific results for related terms. This method will be used to expand the search queries in order to provide broader, narrower or generally related results.

In the following chapter, the scalability and feasibility of the proposed method will be tested using two different unstructured P2P search techniques. The results of experiments will be presented and compared with each other.

Chapter 4

Evaluation and Results

As explained in Chapter 3 in order to tackle the scalability issue of previous approaches, we proposed a distributed model that provides recommendations by performing search over unstructured P2P network obtained from friendship graph of social networking sites. In this chapter two experiments are performed in order to test the scalability of the proposed methodology. Observations and results are presented and analyzed in the further sections.

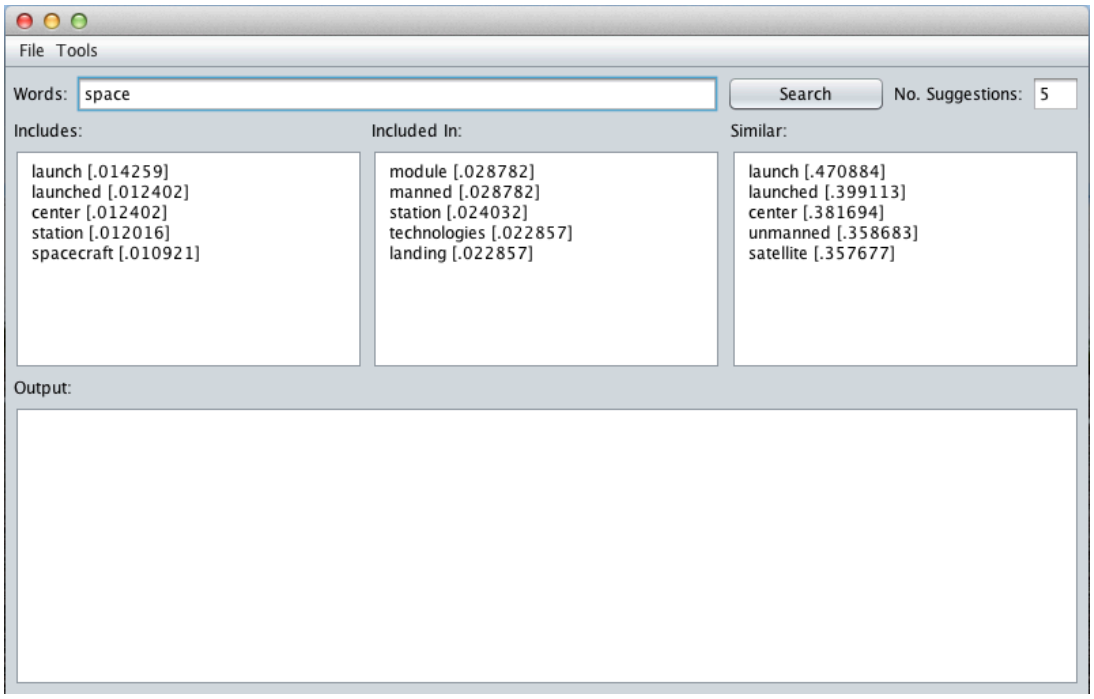
4.1 Developed Recommendation System

The prototype application of the Recommendation System proposed in Chapter 3 based on the fuzzy thesaurus knowledge-base was implemented. The prototype was built based on 30 documents containing 5961 unique words excluding words such as “some”, “what” and “this” which are called stop words. In addition, a P2P network simulation was utilized in order to test the performance and feasibility of the proposed distributed model. In this P2P network it was assumed each node stores its own thesaurus knowledge-base that will be employed by the distributed Recommendation System. Figure 4.1 shows a screen-shot of the developed Recommendation System application, where the user entered the term “space” in the search field. The three different types of suggestions listed here are: Narrower Terms, Broader Terms and Related Terms, which in this application are presented as “Includes”, “Included in” and “Similar” terms respectively. Also this application allows the user to choose the number of suggested terms required where in this example is 5.

As shown in Figure 4.1, the first column on the left displays the top five Narrower Terms for the term “space”. The second column displays the Broader Terms of “space” and the third column displays the list of Related Terms of “space”. All three columns are sorted based on the degree of relationship in descending order.

4.2 Underlying Network Scalability Test

In this section the scalability of the system is studied. Each node in the network has its own knowledge-base, which is constructed based on the documents it contains. The Recommendation System is available to each node and its neighbors while the user can access it when searching for a keyword. As the user types in a keyword in the Recommendation System interface, the distributed search is triggered and that specific node sends a keyword query to all of its neighbors. In cases where the keyword is found in a node (i.e. hit) all related terms with the degree of relation are sent back to the node that initially triggered the search. If the node does not match the query with its knowledge-base it will forward the query to directly connected neighbors according to the search algorithm used. This process continues until the message reaches the maximum number of hops that the query is allowed to travel before it gets removed from the network.



Words: space

No. Suggestions: 5

Includes:	Included In:	Similar:
launch [.014259]	module [.028782]	launch [.470884]
launched [.012402]	manned [.028782]	launched [.399113]
center [.012402]	station [.024032]	center [.381694]
station [.012016]	technologies [.022857]	unmanned [.358683]
spacecraft [.010921]	landing [.022857]	satellite [.357677]

Figure 4.1: *User types in the first word*

The parameters of performance metrics in this study are as follows:

1. Number of query hits: indicates the number of objects successfully found for a search query on the network.

2. Number of sent messages: indicates the number of sent messages on behalf of each search query. The lower the value, the less traffic produced in the network.

3. Success Ratio: represents the efficiency of the search method. For each query this percentage is calculated by dividing the number of query hits by the total number of sent messages.

4. Delay: represents the total amount of time required in *ms* in order to complete a

search query. This value can be obtained from the sum of all *network delays* for the last query hit.

4.3 Experiment 1: Generating The Unique Words Database

In an experiment 300 random documents with approximately the same length were obtained from Wikipedia and added to the Recommendation System's database. The results of the experiment indicate the total number of unique words in 300 documents are equal to 26318. Out of which, 5961 of the total unique words were added to the database by the time the first 30 documents were inserted. Therefore, by selecting 30 documents out of 300 we should have a knowledge-base with approximately 5961 unique terms and their relationships. The statistics of this experiment will be used in both experiments in Sections 4.5.1 and 4.5.2. Although these statistics are not very accurate, yet they provide good approximations, which can be used in simulation environment.

4.4 Simulation's Environment

As mentioned in Chapter 2, *blind search* methods are relatively more practical and convenient in most cases. Hence in this study two major categories of blind search methods in unstructured P2P networks: *Flooding* and *Random Walk* are applied. P2P network simulation were performed using the search framework written on top of PeerSim P2P network simulator program [55]. Both PeerSim and the search framework are written in Java.

Since it is required to generate a graph for social networking sites, the network's

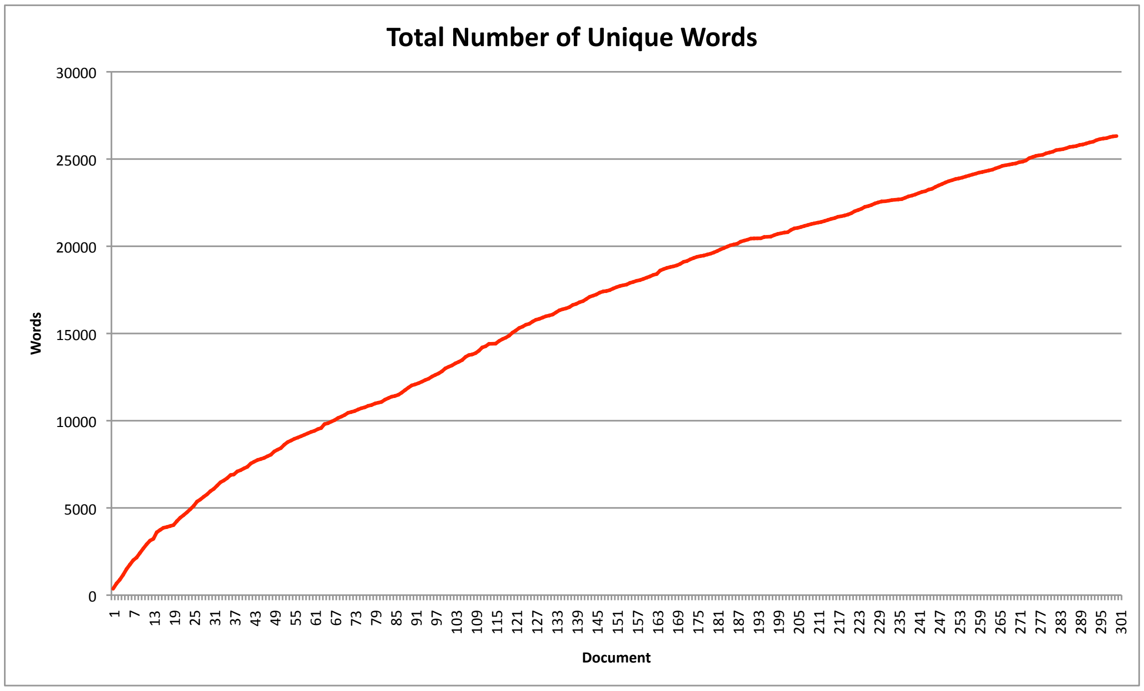


Figure 4.2: *The growth pattern of total number of unique words for 300 documents added to the system*

wiring is set to be random with small-world characteristics. A network with small-world characteristics can potentially increase the accuracy of search results. The reason is that each node can reach any other node in few steps. In this work the small-world model proposed by Watts and Strogatz [56] were used, which has closer properties to existing social networks.

It is assumed that there are 300 documents available on the entire network. These 300 documents contain 26318 unique words. Also, each node is assumed to contain 30 documents. According to the growth pattern of the total number of unique words in Figure 4.2 in Section 4.3, 30 documents approximately contain 5961 unique words. Thus it can be assumed that each node contains about 5961 words out of the total 26318 unique words available on the system. However, since the set of 5961 words in each node is a subset of the 26318 unique words on the entire system, thus different nodes may have a number of words in common.

In order to avoid the randomness of the results, the network is set to perform 50 queries for each simulation run. In both experiments the following network configuration were applied:

- Total number of unique words in the entire system: 26318
- Maximum number of connections per node: 20
- Total number of unique words each node contains: 5961
- TTL: 4
- Minimum network delay: 50
- Maximum network delay : 400
- Number of queries: 50

Searching over unstructured P2P networks generates a huge traffic upon each search request [57]. Specially in the *flooding* technique, it is tried to reasonably set the TTL value to its minimum. This value indicates the number of hops a message travels before it gets removed from the system. Since the generated graph is supposed to have small-world characteristics, the TTL value is set to 4.

The network delay for each message is considered to be a random value between 50-400ms. Network delay is an important performance characteristics of computer networks. This value represents the amount of time it takes for a packet to travel from one node to another [58].

4.5 Simulation Results

The simulations were only performed in non-dynamic environment, which means that the number of nodes from the start to the end of the simulation were fixed. In order to test the scalability of the proposed system, in each experiment three simulation runs of different network sizes were performed. The network size value indicates the total number of nodes in the network.

In the first run of each experiment the network size or the total number of nodes in the network is set to 750 nodes. The second and third runs were set to 2500 and 5000 nodes respectively. All other parameters remain the same as previously stated.

4.5.1 Experiment 2: Random Walk Search Method

Experiment 2 is performed based on the *Random Walk* technique. In this experiment as the search is triggered, a random walker containing the searched keyword is sent to each neighboring node, which makes the k value equal to the number of connections a node has. In the case that any neighbor matches the query to its local knowledge-base, it will notify the initiator of the search query and reports the query as *hit*. Thus the random walker is stopped. Alternatively, if the neighboring node does not match the query with its knowledge-base it will forward the random walker to one of its random neighbors. This process will carry on for each random walker until it either succeeds or reaches the TTL value.

Figures 4.3, 4.4 and 4.5 show the three runs in the experiment 2. The searched keywords in all 50 queries were successfully found. In the first run with a network containing 750 nodes, an average rate of 17.10 hits per query was achieved. This value is almost the same as the second and third runs with the average rate of 17.38 and 17.09

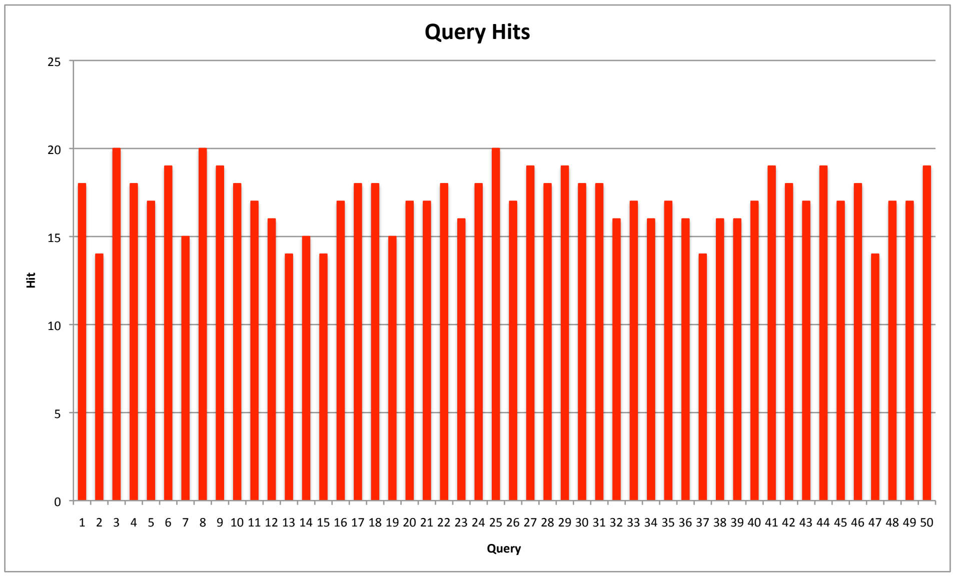


Figure 4.3: *Random Walk Method - The number of hits for each query in a network with 750 nodes*

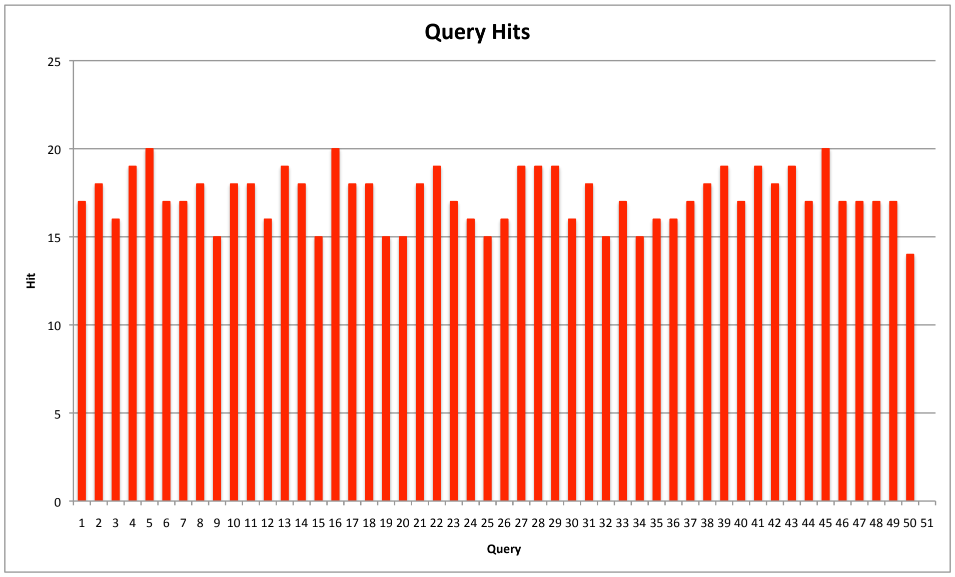


Figure 4.4: *Random Walk Method - The number of hits for each query in a network with 2500 nodes*

hits respectively.

Measuring the amount of messages transfered on behalf of each sent query indicates

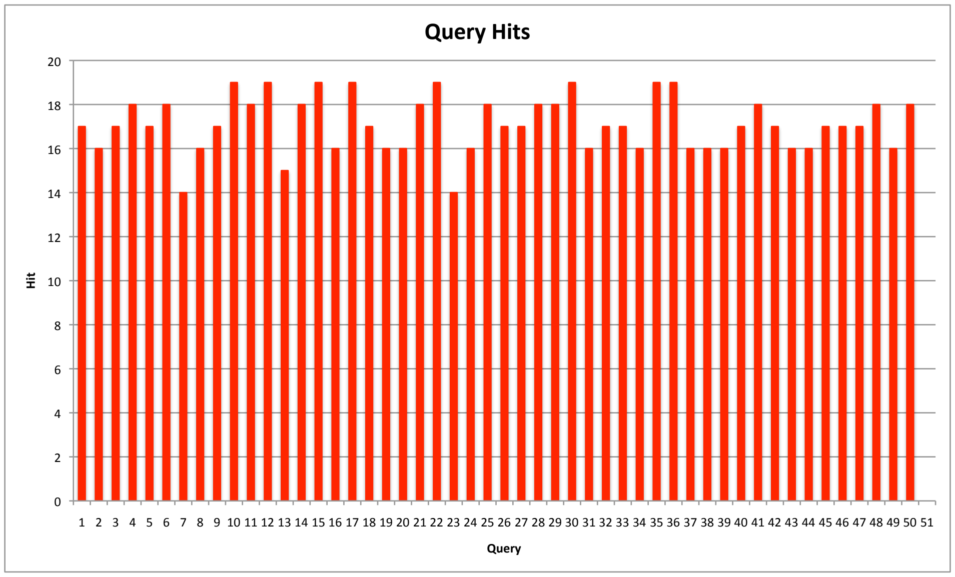


Figure 4.5: *Random Walk Method - The number of hits for each query in a network with 5000 nodes*

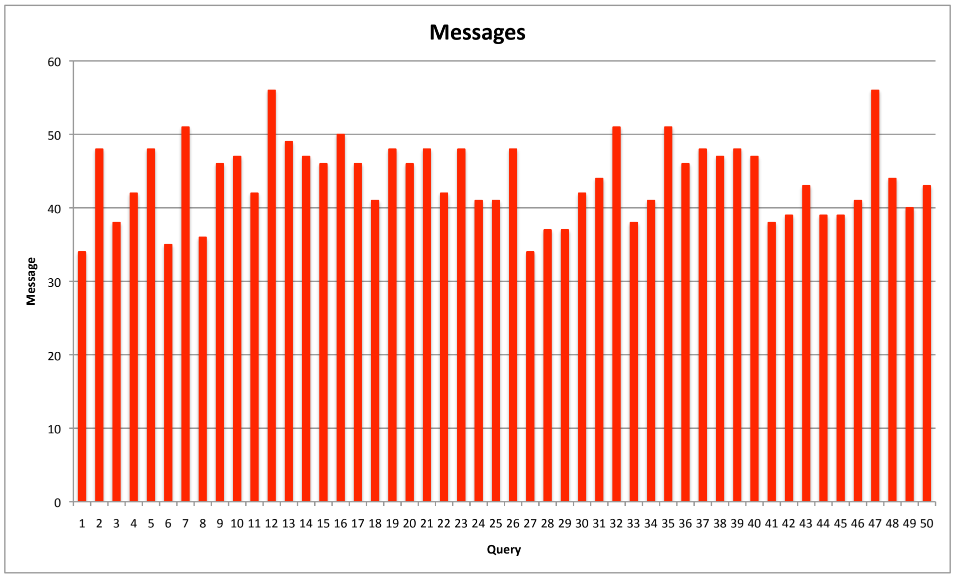


Figure 4.6: *Random Walk Method - The total number of messages sent on behalf of each query in a network with 750 nodes*

the network traffic. In all three network sizes similar results were observed. Figures 4.6, 4.7 and 4.8 show the number of transfered messages for each query sent in each

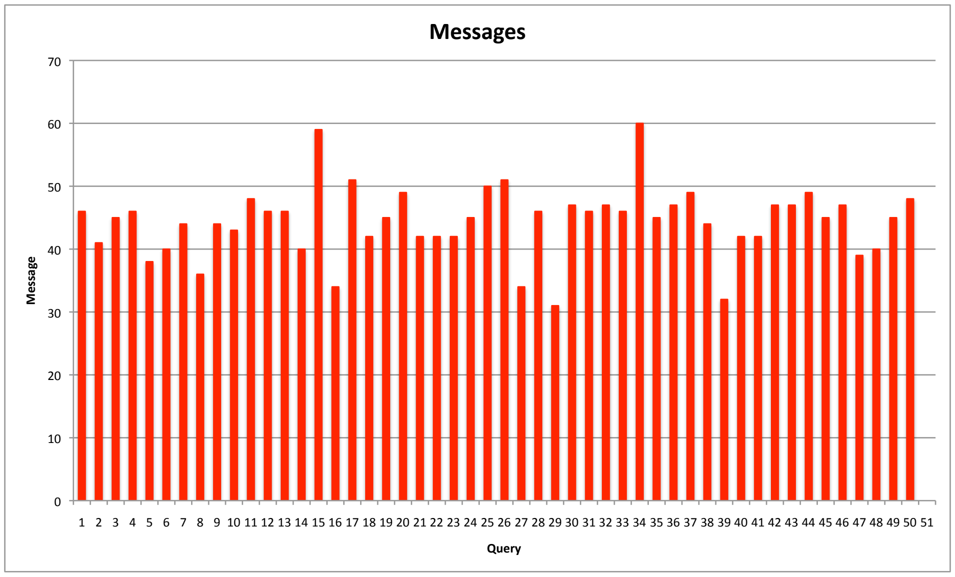


Figure 4.7: *Random Walk Method - The total number of messages sent on behalf of each query in a network with 2500 nodes*

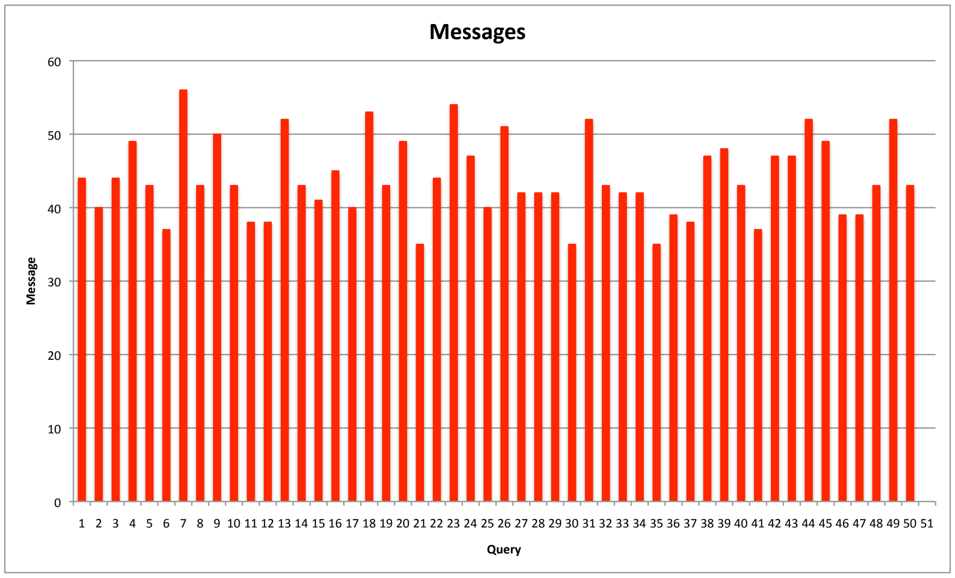


Figure 4.8: *Random Walk Method - The total number of messages sent on behalf of each query in a network with 5000 nodes*

network. The average number of messages produced for each query in the first run was 43.91 messages. Similarly, this value is almost the same as the other two average values of 44.32 and 43.96 messages for the second and the third runs.

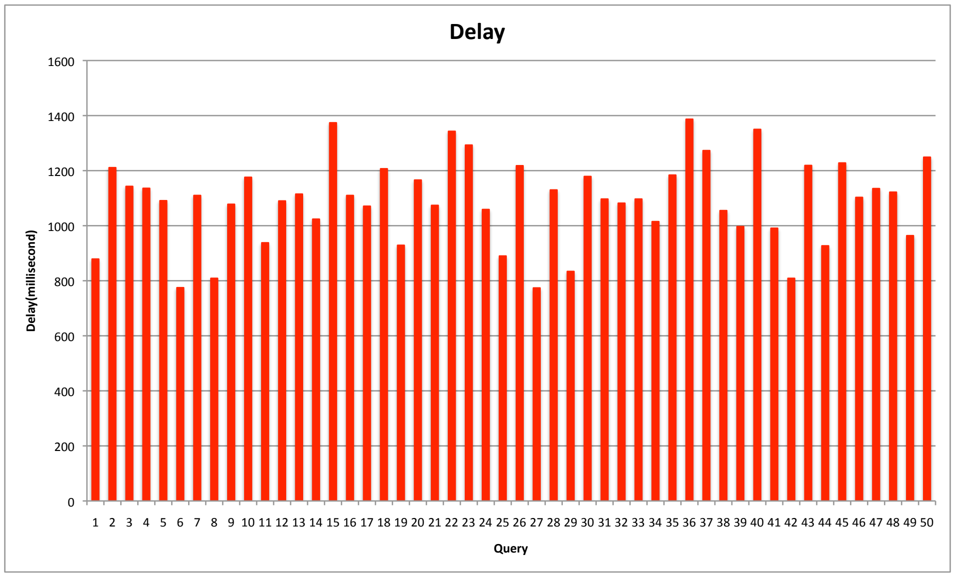


Figure 4.9: *Random Walk Method - The maximum delay time for each query in a network with 750 nodes*

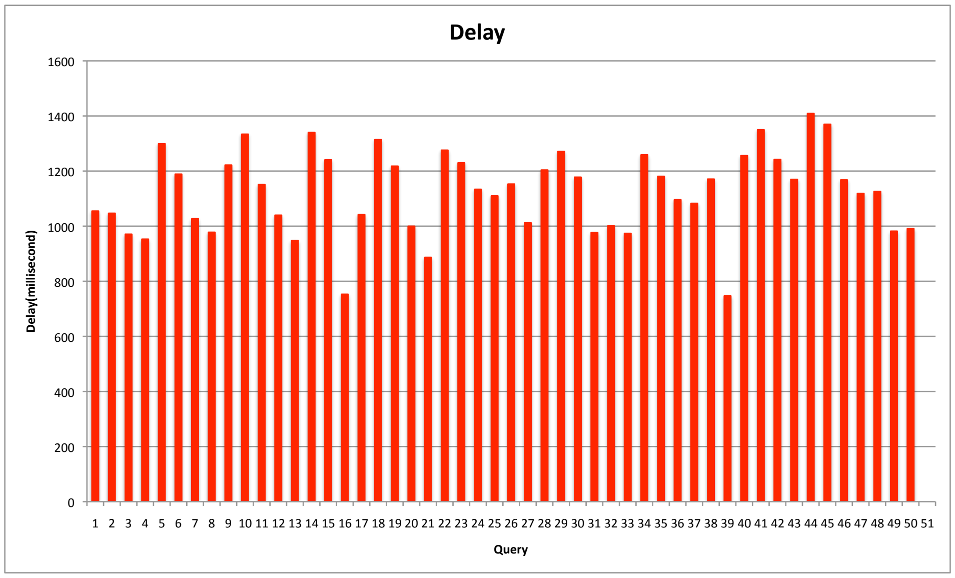


Figure 4.10: *Random Walk Method - The maximum delay time for each query in a network with 2500 nodes*

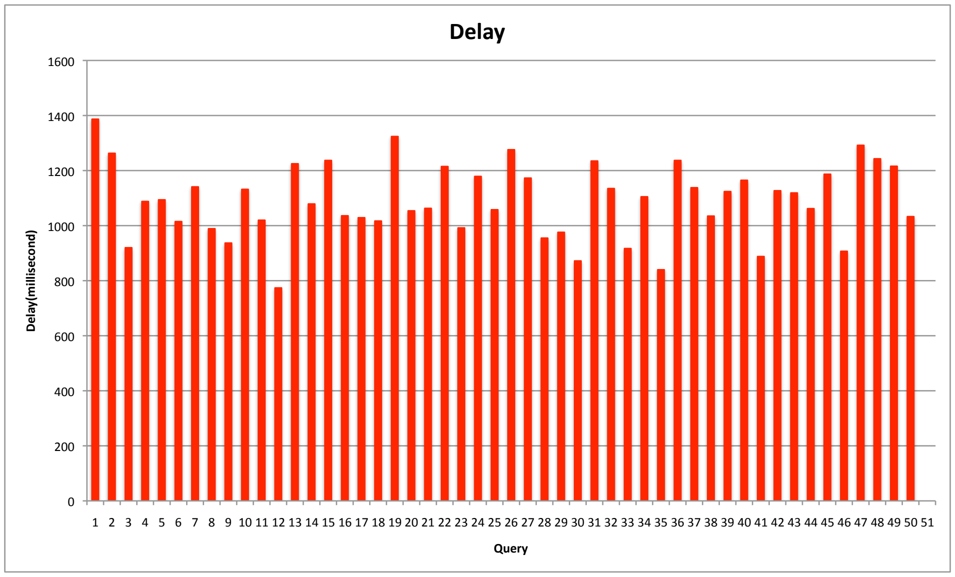


Figure 4.11: *Random Walk Method - The maximum delay time for each query in a network with 5000 nodes*

Figures 4.9, 4.10 and 4.11 show the maximum amount of time that each query took in order to obtain all results from nodes which reported *hit* for the searched keyword. The maximum of 1087ms for the first, 1128ms for the second and 1089ms for the third runs were observed. The results imply that completing a query in all three network sizes takes about the same amount of time. Therefore the delay time is independent from size of the network.

4.5.2 Experiment 3: Flooding Method

Experiment 3 is performed based on Gnutella like *Flooding* technique. The steps in this approach are the same as the steps in the *Random Walk* technique used in experiment 2. The two methods differ in a crucial step. Unlike the *Random Walk* technique, where nodes forward the message to one randomly selected neighbor, in the *Flooding* technique, nodes forward the message to all their neighbors.

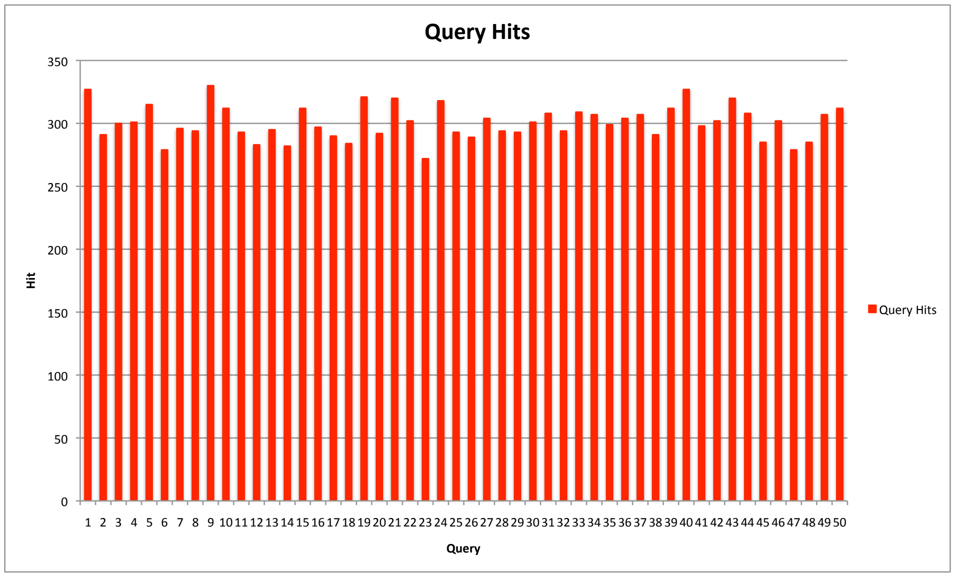


Figure 4.12: *Flooding Method - The number of hits for each query in a network with 750 nodes*

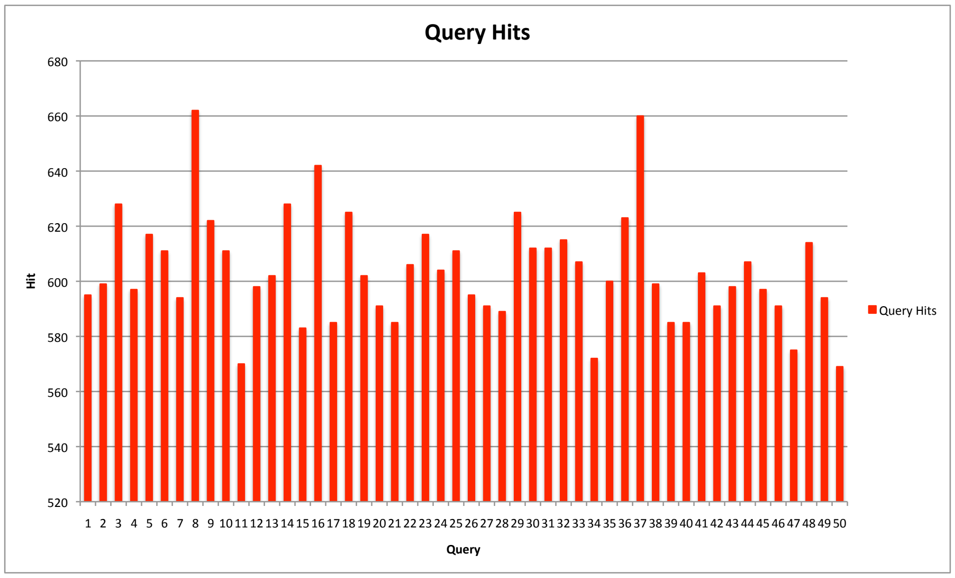


Figure 4.13: *Flooding Method - The number of hits for each query in a network with 2500 nodes*

The three runs of the experiment 3 are shown in Figures 4.12, 4.13 and 4.14. Similar to the results achieved in experiment 2, all of the 50 searched queries had at least 1 result.

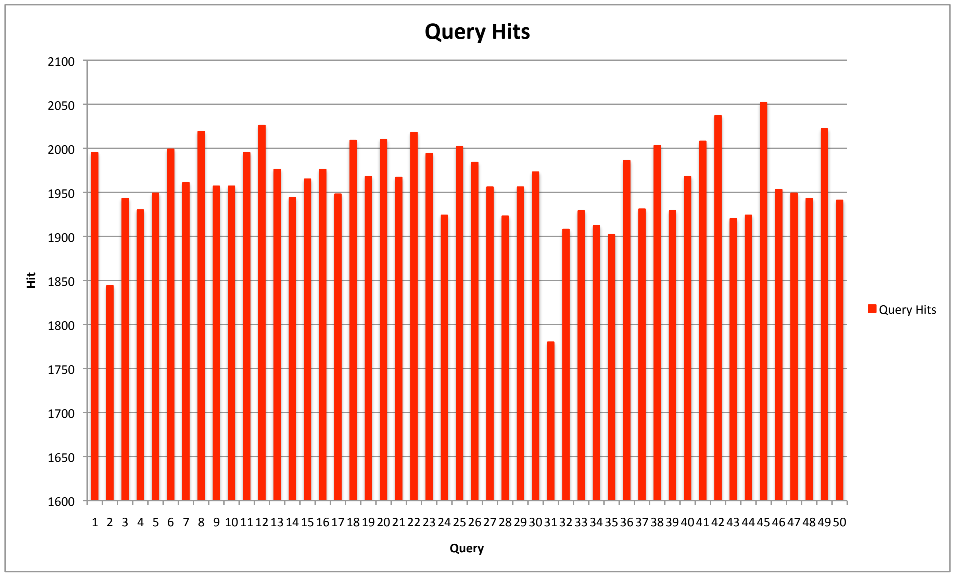


Figure 4.14: *Flooding Method - The number of hits for each query in a network with 5000 nodes*

However, unlike the experiment 2, the results of the average hit rate in the three runs are not close. In the first trial where the number of nodes were 750, an average rate of 300.64 hits per query was observed. While these values for the network size of 2500 and 5000 nodes were 603.49 and 1960.68 hits respectively. Thus, as the number of nodes in network increases the *flooding* technique provides higher hit rates.

Figures 4.15, 4.16 and 4.17 represent the network traffic based on the number of messages sent in each run. The average value obtained for the network size of 750 nodes is 35400.31 messages where this value for 2500 and 5000 nodes is 35849.73 and 35414.01 messages respectively. Again as shown in Table 4.2 the average values of sent messages are within the same range.

Figures 4.18, 4.19 and 4.20 show the maximum amount of time that each query takes in order to obtain all results. In first run the maximum delay time is 1394.50ms where in the second and the third runs these values are 1437.15ms and 1496.94ms respectively.

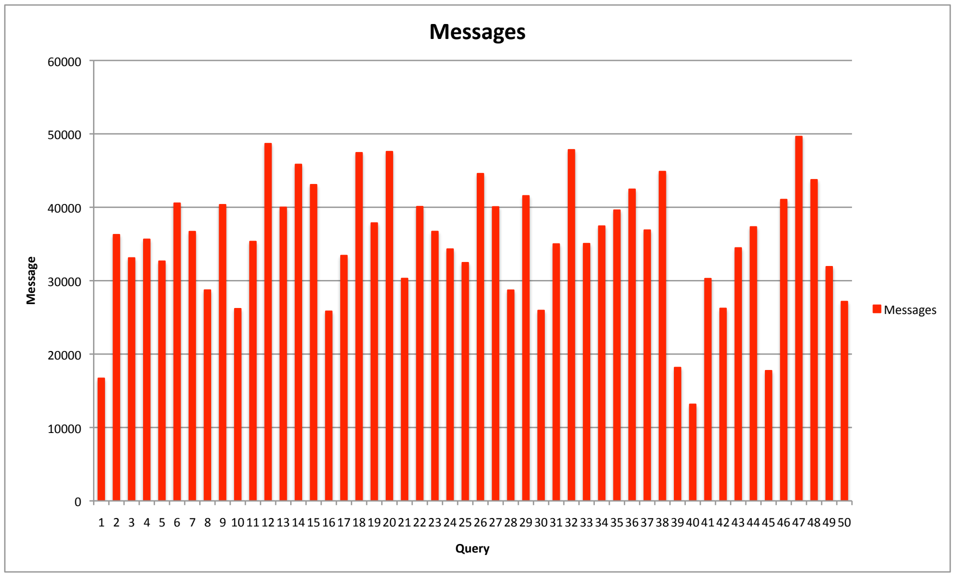


Figure 4.15: *Flooding Method - The total number of messages sent on behalf of each query in a network with 750 nodes*

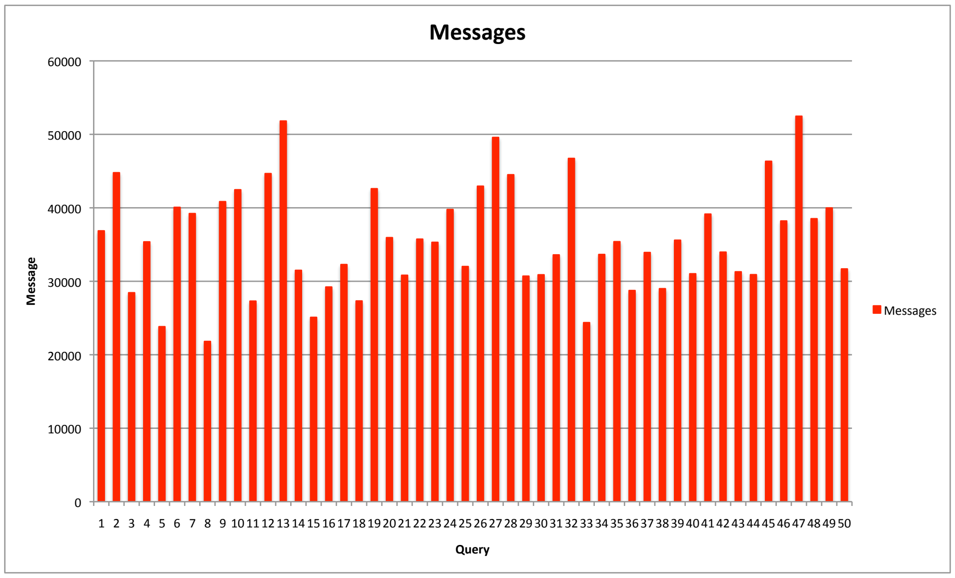


Figure 4.16: *Flooding Method - The total number of messages sent on behalf of each query in a network with 2500 nodes*

Although the number of nodes in the network is increasing, the maximum delay times are within the same range.

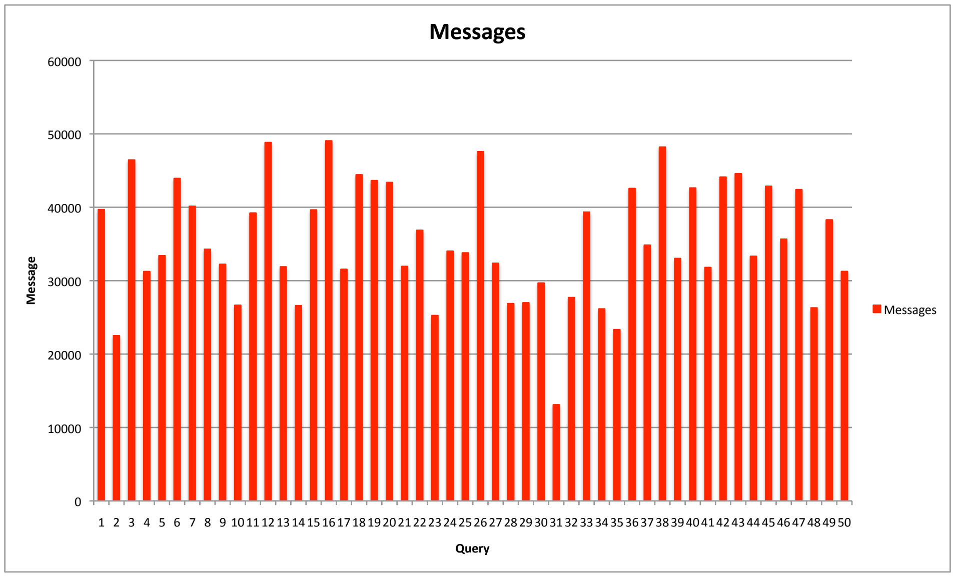


Figure 4.17: *Flooding Method - The total number of messages sent on behalf of each query in a network with 5000 nodes*

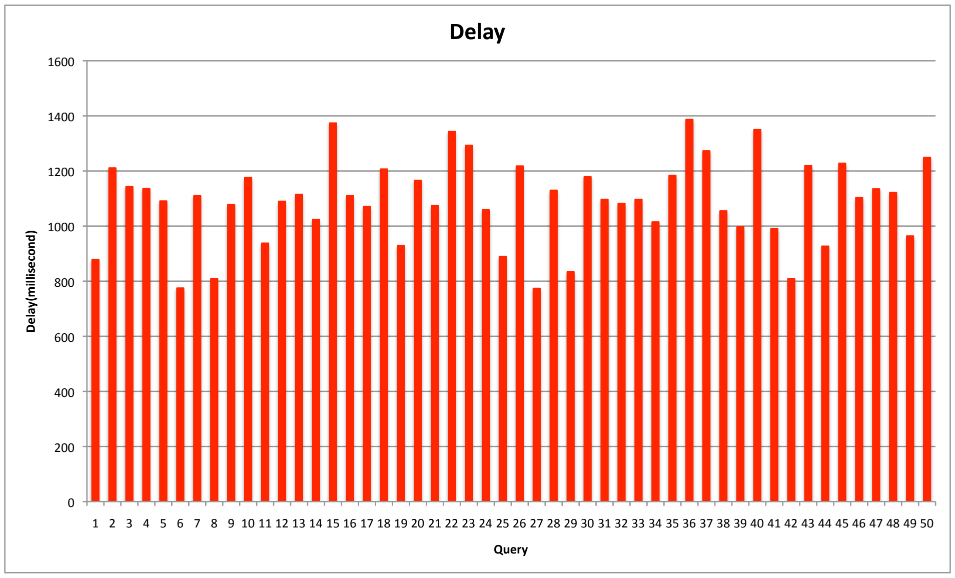


Figure 4.18: *Flooding Method - The maximum delay time for each query in a network with 750 nodes*

4.6 Random Walk vs. Flooding Technique

The *success ratio* is calculated based on finding the ratio between the *number of query hits* and the *number of sent messages.* The higher percentage of *success ratio* indicates

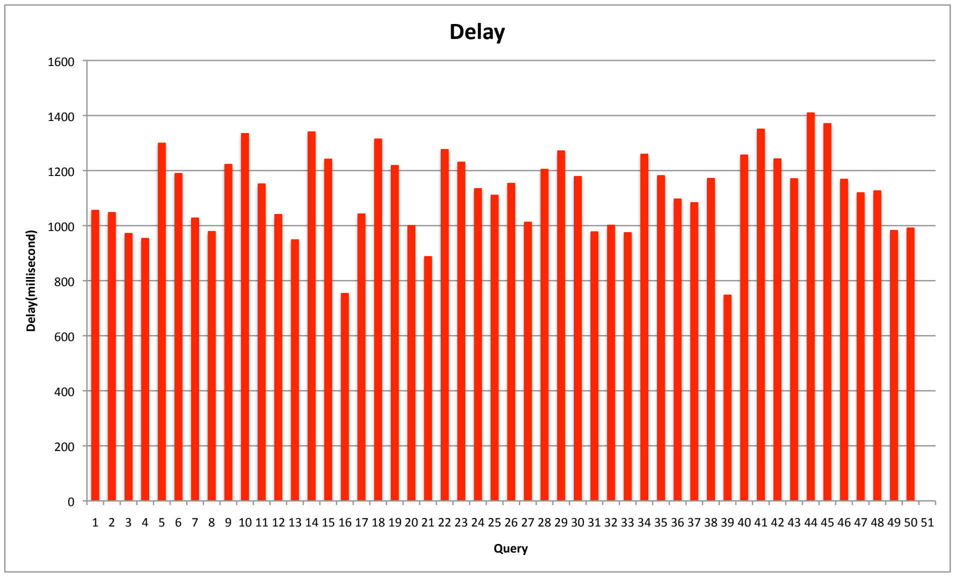


Figure 4.19: *Flooding Method - The maximum delay time for each query in a network with 2500 nodes*

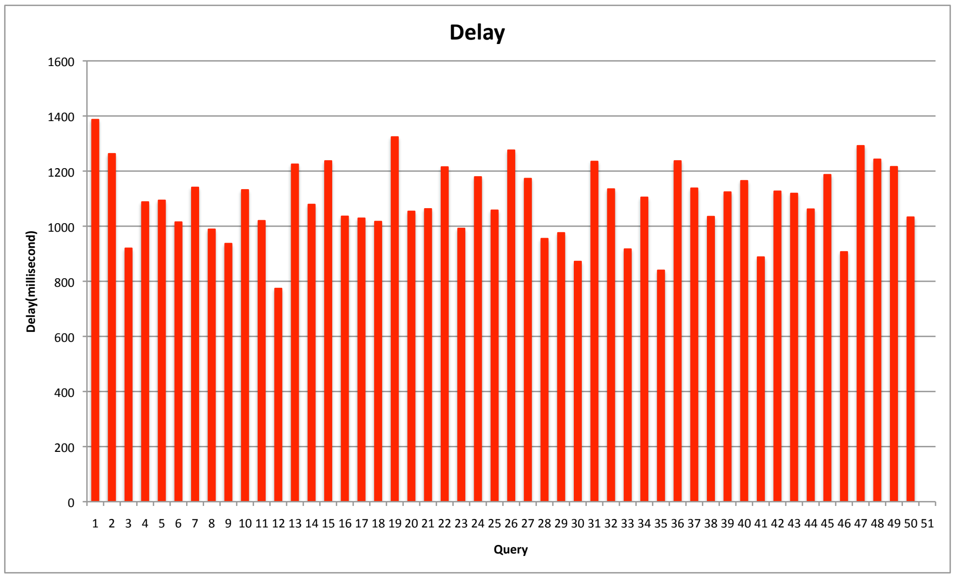


Figure 4.20: *Flooding Method - The maximum delay time for each query in a network with 5000 nodes*

the higher efficiency of the search method. For instance in Table 4.1, the success ratio of the three runs are 38.94%, 38.99% and 38.88%. These values indicate that the success rate of the *Random Walk* technique is independent of the network size.

In contrast to the experiment 2, Table 4.2 indicates that the *success ratio* of *Flooding* technique is dependent on the size of the network. The larger the size of network is, the higher the *success ratio*. For instance in the runs one, two and three of the *Flooding* technique, the *success ratio* are 0.85%, 1.68% and 5.54%. These values are not within the same range. In comparison to experiment 2 where the *success ratio* is relatively constant, in experiment 3 this ratio is dramatically increasing as network size grows. Yet, the generated traffic over the network is also increasing.

Table 4.1: *Results for the Random Walk experiment*

No. Nodes	Avg. Hit Rate	Max. Delay	Avg. No. Messages	Success Ratio
750	17.10	1087.38 *ms*	43.91	38.94 %
2500	17.28	1128.06 *ms*	44.32	38.99 %
5000	17.09	1089.00 *ms*	43.96	38.88 %

Table 4.2: *Results for the Flooding experiment*

No. Nodes	Avg. Hit Rate	Max. Delay	Avg. No. Messages	Success Ratio
750	300.64	1394.50 *ms*	35400.31	0.85 %
2500	603.49	1437.15 *ms*	35849.73	1.68 %
5000	1960.68	1496.94 *ms*	35414.01	5.54 %

Although having more hits tends to produce more accurate results for the Recommendation System; nevertheless, by comparing the generated traffic and the *success ratio* of two experiments, it can be suggested that the *Random Walk* technique provides a better solution. Therefore, we suggest a searching method based on *Random Walk* for the fuzzy recommendation system.

4.7 Summary

In this chapter, two different types of blind unstructured P2P search techniques have been examined in order to test the scalability of proposed distributed search method in unstructured P2P networks. Results of the *Random Walk* and *Flooding* search techniques were shown and compared with each other. As demonstrated, results suggest that the performance of the proposed Recommendation System is reasonable and scalable at least up to 5000 nodes when it uses a *Random Walk* search technique in the underlying network. Therefore it can be employed in real networks.

Chapter 5

Conclusion and Future Work

In this section a summary of the study in this thesis is presented and the plan for future work is discussed.

5.1 Conclusion

In this study a distributed Recommendation System is proposed that can be used for query expansion in Web2.0 social networking web sites. The proposed Recommendation System uses a fuzzy-based model in order to guide users by employing the knowledge-bases of related words. The backbone of the system is an unstructured P2P network of nodes connected to one another. In each node the knowledge-base data is stored and updated by adding documents. This network extends the search capability by performing distributed search over all knowledge-bases stored on other neighboring nodes. The real example of this P2P network can be Web2.0 social networking sites or the networks consisting of digital libraries. A developed prototype of the Recommendation System and the simulation results show the scalability and feasibility of this system.

As simulation was performed and results were analyzed, the amount of traffic produced from the messages in two famous search techniques: *Random Walk* and *Flooding* were studied. The results indicate the *Random Walk* approach is better considering the *query hit ratio* and the amount of the generated traffic. In addition, the scalability test was performed by increasing the number of nodes from 750 to 5000 nodes. The achieved results indicate that the increase in the network size does not have significant affect on the network traffic and the delay in delivering results. Also the average delay show that the amount of time that each query take in order to deliver the results is in a reasonable range. Therefore results show the proposed system is scalable since the network traffic increase is marginal.

5.2 Future Work

The future direction for this research is enhancing the Recommendation System to employ novel data-mining approaches for information retrieval and processing of the data that contains uncertain information.

For example one direction is the system will also be capable of providing summarized information that is based on the correct interpretation of perception-based data existing in Web2.0 sites.

Another direction is trying to find accurate data from inaccurate information of Web2.0 sites. For example developing methods to be able to identify the users in different Web2.0 social network sites are the future direction of this research. This will help the Recommendation System improve its results by using extra knowledge-bases.

Also enhancing this search engine to consider the attributes of XML document in the semantic web is another direction for this work.

This research can also provide insights in the area of information retrieval that will contribute to the evolution of search engines for semantic web (i.e., Web3.0.) in the future.

References

[1] T. Ghasemi and A. Abhari, “A distributed fuzzy recommendation system,” in *Proceedings of the 2012 Spring Simulation Multiconference*, SpringSim ’12, (San Diego, CA, USA), Society for Computer Simulation International, 2012.

[2] A. Singhal, “Modern Information Retrieval: A Brief Overview,” *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol. 24, no. 4, pp. 35–42, 2001.

[3] G. McMurdo, “How the internet was indexed,” vol. 21, pp. 479–489, Journal of Information Science, 1995.

[4] S. K. T. Seymour, D. Frantsvog, “History of search engines,” *International Journal Of Management and Information Systems*, vol. 15, no. 4, 2011.

[5] “Search Engine Market Share.” http://www.netmarketshare.com/search-engine-market-share.aspx, Last visited on March 2012.

[6] C. Kousetti, D. E. Millard, and Y. Howard, “A study of ontology convergence in a semantic wiki,” in *Proceedings of the 4th International Symposium on Wikis*, WikiSym ’08, (New York, NY, USA), pp. 17:1–17:10, ACM, 2008.

[7] "List of Wikipedias." http://meta.wikimedia.org/wiki/List_of_Wikipedias, Last visited on March 2012.

[8] S. G. Martine De Cock and M. Nikravesh, *Fuzzy Thesauri for and from the WWW*, vol. 164 of *Studies in Fuzziness and Soft Computing*, pp. 275–284. New York: Springer-Verlag, 2005.

[9] L. Zadeh, "Fuzzy sets," vol. 8, pp. 338–353, Information and Control, 1965.

[10] A. Bargiela and W. Pedrycz, *Granular Computing: An Introduction (Springer International Series in Engineering and Computer Science).* Springer, Nov. 2002.

[11] S. Shang, Y.-L. Wu, and O. Hou, "An analysis of business models of web 2.0 application," in *Information Technology: New Generations, 2009. ITNG 09. Sixth International Conference on*, pp. 314 –319, 2009.

[12] J. Musser, *Web 2.0 Principles and Best Practices.* O Reilly Media, Inc, 2006.

[13] S. Murugesan, "Understanding web 2.0," vol. 9, pp. 34–41, 2007.

[14] C. Oswals and K. Nilson, "Web 2.0 fundamentals for developers: With AJAX, development tools, and mobile platforms," pp. 235 –236, Jones and Bartlett Learning, 2010.

[15] S. Milgram, "The small world problem," *Psychology Today*, vol. 61, pp. 60–67, 1967.

[16] S. Pool and M. Kochen, "Contacts and influence," vol. 1, pp. 5–51, Social Networks, 1978.

[17] C. Li-mei, “Micro-blog social network analysis: ”digu” network as an example,” pp. 1–3, Information Engineering and Electronic Commerce (IEEC), 2010 2nd International Symposium on, 2010.

[18] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, “Measurement and analysis of online social networks,” in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC ’07, (New York, NY, USA), pp. 29–42, ACM, 2007.

[19] A. S. Tanenbaum and M. V. Steen, “Distributed systems, principles and paradigms,” pp. 2–3, Pearson Prentice Hall, 2007.

[20] I. Stoica, R. Morris, D. Karger, F. M. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, vol. 31, (New York, NY, USA), pp. 149–160, ACM, October 2001.

[21] H. N. Chan, K. N. Van, and G. N. Hoang, “Characterizing Chord, Kelips and Tapestry algorithms in P2P streaming applications over wireless network,” IEEE, 2008.

[22] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, “Making gnutella-like P2P systems scalable,” in *SIGCOMM ’03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 407–418, ACM, 2003.

[23] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, “A local search mechanism for peer-to-peer networks,” in *Proceedings of the eleventh international conference*

on Information and knowledge management, CIKM '02, (New York, NY, USA), pp. 300–307, ACM, 2002.

[24] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of the 16th international conference on Supercomputing*, ICS '02, (New York, NY, USA), pp. 84–95, ACM, 2002.

[25] L. Rodero-Merino, A. F. Anta, L. López, and V. Cholvi, "Self-managed topologies in P2P networks," *Computer Networks*, vol. 53, pp. 1722–1736, July 2009.

[26] D. Tsoumakos and N. Roussopoulos, "Analysis and comparison of P2P search methods," in *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems*, (New York, NY, USA), p. 25, ACM, 2006.

[27] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," in *Proceedings of the eleventh international conference on Information and knowledge management*, CIKM '02, (New York, NY, USA), pp. 300–307, ACM, 2002.

[28] D. A. Menascé and L. Kanchanapalli, "Probabilistic scalable P2P resource location services," *SIGMETRICS Perform. Eval. Rev.*, vol. 30, pp. 48–58, september 2002.

[29] H. S. Christopher D. Manning, Prabhakar Raghavan, *An Introduction to Information Retrieval.* Cambridge University Press, 2009.

[30] D. Walker, "Query expansion using thesauri: Previous approaches and possible new directions," tech. rep., University of California, Los Angeles, 405 Hilgard Ave, Los Angeles, California, USA, 2001.

[31] "Yahoo Search Engine." http://www.yahoo.com, Last visited on March 2012.

[32] "Unified Medical Language System (UMLS)." http://www.nlm.nih.gov/research/umls/, Last visited on March 2012.

[33] H. Schutze and J. O. Pedersen, "A cooccurrence-based thesaurus and two applications to information retrieval," *Information Processing and Management*, vol. 33, no. 3, pp. 307 – 318, 1997.

[34] B. P. McCune, R. M. Tong, J. S. Dean, and D. G. Shapiro, "Rubric: A system for rule-based information retrieval," *IEEE Transactions on Software Engineering*, vol. 11, pp. 939–945, September 1985.

[35] F. Pereira, N. Tishby, and L. Lee, "Distributional clustering of english words," in *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, ACL '93, (Stroudsburg, PA, USA), pp. 183–190, Association for Computational Linguistics, 1993.

[36] G. Ruge, "Automatic detection of thesaurus relations for information retrieval applications," in *Foundations of Computer Science: Potential - Theory - Cognition, to Wilfried Brauer on the occasion of his sixtieth birthday*, (London, UK, UK), pp. 499–506, Springer-Verlag, 1997.

[37] D. Lin, "Automatic retrieval and clustering of similar words," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, (Stroudsburg, PA, USA), pp. 768–774, Association for Computational Linguistics, 1998.

[38] M. Sanderson and B. Croft, "Deriving concept hierarchies from text," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and develop-*

ment in information retrieval, SIGIR '99, (New York, NY, USA), pp. 206–213, ACM, 1999.

[39] B. R.-N. Ricardo Baeza-Yates, *Modern Information Retrieval.* Adison Wesley, 1999.

[40] W. B. Frakes and R. A. Baeza-Yates, eds., *Information Retrieval: Data Structures & Algorithms.* Prentice-Hall, 1992.

[41] Y.-J. Horng, S.-M. Chen, Y.-C. Chang, and C.-H. Lee, "A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques," *Fuzzy Systems, IEEE Transactions on*, vol. 13, pp. 216 – 228, april 2005.

[42] H. Lee and J. Kwon, "Improving context awareness information retrieval with online social networks," in *Proceedings of the 2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, CNSI '11, (Washington, DC, USA), pp. 391–395, IEEE Computer Society, 2011.

[43] J. He, *A social network-based recommender system.* PhD thesis, Los Angeles, CA, USA, 2010. AAI3437557.

[44] G. Pasi, "Fuzzy sets in information retrieval: State of the art and research trends," in *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models* (H. Bustince, F. Herrera, and J. Montero, eds.), vol. 220 of *Studies in Fuzziness and Soft Computing*, pp. 517–535, Springer Berlin / Heidelberg, 2008.

[45] D. H. Kraft, G. Pasi, and G. Bordogna, "Vagueness and uncertainty in information retrieval: how can fuzzy sets help?," in *Proceedings of the 2006 international workshop on Research issues in digital libraries*, IWRIDL '06, (New York, NY, USA), pp. 3:1–3:10, ACM, 2007.

[46] L. Zadeh, “From search engines to question-answering systems: The problems of world knowledge, relevance, deduction, and precisiation,” in *FUZZY LOGIC AND THE SEMANTIC WEB* (E. Sanchez, ed.), Elsevier B.V., 2006.

[47] G. Korotkikh, *A New Fuzzy Spectral Approach to Information Integration in a Search Engine*, vol. 164 of *Studies in Fuzziness and Soft Computing*, pp. 327–338. Springer Berlin / Heidelberg, 2005.

[48] S. Miyamoto, “Information retrieval based on fuzzy associations,” *Fuzzy Sets Syst.*, vol. 38, pp. 191–205, november 1990.

[49] K. Nomoto, S. Wakayama, T. Kirimoto, Y. Ohashi, and M. Kondo, “A document retrieval system based on citations using fuzzy graphs,” *Fuzzy Sets Syst.*, vol. 38, pp. 207–222, november 1990.

[50] M. Beigbeder and A. Mercier, “An information retrieval model using the fuzzy proximity degree of term occurences,” in *Proceedings of the 2005 ACM symposium on Applied computing*, SAC ’05, (New York, NY, USA), pp. 1018–1022, ACM, 2005.

[51] D. Parry, “A fuzzy ontology for medical document retrieval,” in *Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation - Volume 32*, ACSW Frontiers ’04, (Darlinghurst, Australia, Australia), pp. 121–126, Australian Computer Society, Inc., 2004.

[52] C. A. Bean and R. Green, *Relationships in the organization of knowledge*, vol. 2 of *Information Science and Knowledge Management*. Springer, 2001.

[53] M. W. Berry and M. Browne, *Understanding Search Engines.* Philadephia, PA: Society for Industrial and Applied Mathematics, 2005.

[54] W. Zeng and H. Li, “Inclusion measures, similarity measures, and the fuzziness of fuzzy sets and their relations: Research articles,” *Int. J. Intell. Syst.*, vol. 21, pp. 639–653, june 2006.

[55] A. Montresor and M. Jelasity, “PeerSim: A scalable P2P simulator,” in *Proceeding of the 9th Int. Conference on Peer-to-Peer (P2P’09)*, (Seattle, WA), pp. 99–100, 2009.

[56] D. J. Watts and S. H. Strogatz, “Collective dynamics of ’small-world’ networks,” vol. 393, (Department of Theoretical and Applied Mechanics, Cornell University, Ithaca, New York 14853, USA. djw24@columbia.edu), pp. 440–442, Nature Publishing Group, June 1998.

[57] D. Tsoumakos and N. Roussopoulos, “Analysis and comparison of p2p search methods,” in *InfoScale ’06: Proceedings of the 1st international conference on Scalable information systems*, (New York, NY, USA), p. 25, ACM, 2006.

[58] B. Zhang, T. S. E. Ng, A. Nandi, R. Riedi, P. Druschel, and G. Wang, “Measurement based analysis, modeling, and synthesis of the internet delay space,” in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC ’06, (New York, NY, USA), pp. 85–98, ACM, 2006.