RECOMMENDER SYSTEMS – INTEREST GRAPH COMPUTATIONAL METHODS FOR DOCUMENT NETWORKS

by

Gary G. Roberson

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computational Sciences and Informatics

Committee:

_____ Dr. Kirk Borne, Dissertation Director

_____ Dr. Edward Wegman, Committee Chairman

_____ Dr. Larry Kerschberg, Committee Member

_____ Dr. Igor Griva, Committee Member

_____ ___ Dr. Kevin Curtin, Acting Department Chair

_____ Dr. Donna M. Fox, Associate Dean, Office of Student Affairs & Special Programs, College of Science

_____ Dr. Peggy Agouris, Dean, College of Science

Date: _____ Spring Semester 2016
George Mason University
Fairfax, VA

Recommender Systems – Interest Graph Computational Methods for Document Networks

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Gary G. Roberson
Master of Science
George Mason University, 2013

Director: Kirk Borne, Professor
Department of Computational Sciences and Informatics

Spring Semester 2016
George Mason University
Fairfax, VA

DEDICATION

This is dedicated to my two wonderful children, Efrem and Nathan, as motivation to keep challenging themselves their entire lives.

ACKNOWLEDGEMENTS

I would like to thank the many friends, relatives, and supporters who have made this happen. My wife Cynthia who has supported me in whatever challenge I try and my sons who constantly challenge me with new technology they have learned to love.

Dr. Kirk Borne and Dr. Larry Kerschberg have agreed to serve as co-advisors on this research project which does blend computational sciences for big-data with the semantic web to achieve new and valuable approaches for recommender systems. I have found in my class work at GMU an attraction to the classes that both Dr. Kerschberg and Dr. Borne were providing to the Computational Science and Informatics program. So, it was an easy choice for me to approach them to be co-advisors for this research. And, I greatly appreciate their acceptance of my requests and their support in moving this project forward.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

Document Network .....DN

ABSTRACT

RECOMMENDER SYSTEMS – INTEREST GRAPH COMPUTATIONAL METHODS FOR DOCUMENT NETWORKS

Gary G. Roberson, Ph.D.

George Mason University, 2016

Dissertation Director: Dr. Kirk Borne

Recommender Systems are now available in a number of online locations to help users find the reference information they need quicker and with greater accuracy. Document Networks are candidates for this technology to help researchers find research information which pertain to subjects in which they have an interest. Document networks are Bibliographic databases containing scientific publications, preprints, internal reports, as well as databases of datasets used in scientific endeavors such as the World Wide Web (WWW), Digital Libraries, or Scientific Databases (Medline). This Dissertation looks in detail at Document Networks and has chosen Semantic Medline for its case study. Semantic Medline supports thousands of medical researchers who wish to find available citations which pertain to a specific research interest from over 20 million medical research publications. I review Semantic Medline in some detail as well as Recommender Systems and how these systems are constructed and

evaluated. So, the hypothesis is these new approaches will improve Document Network recommendations once implemented. The Dissertation first defines the requirements to improve Document Network recommendations. It then evaluates a host of algorithmic and technical approaches to the problem, selects the best candidate approaches, and a technical platform for evaluation is built to test these optional approaches using the actual Semantic Medline database loaded on a graph database engine. The original Semantic Medline is implemented with a more traditional database approach using MySQL queries to access and bring forward citations for search scenarios. This Dissertation uses new graph tools from social network technology to do the same thing and to evaluate these improved approaches to improve the recommendation accuracy and novelty. After a number of alternative approaches are tried, re-tested, and optimized, the best of the algorithms optimized for Document Networks are found and the original hypothesis is proven while also meeting the requirements. The results are interesting and can lead to greatly improved capabilities for Semantic Medline and for Document Networks in general.

CHAPTER ONE

Introduction

A Recommender System is an information system service that seeks to predict the 'rating' or 'preference' that a user would give to an item of information. For example, a medical researcher needs research information on diseases associated with a specific gene. The Recommender System predicts research references which pertain to this gene. So, the term ‘Recommender Systems’ is a broad area within automated information systems which refers to systems which provide a service based on an explicit, implicit, or inferred knowledge regarding the user of the service. And, they offer recommendations which help the user to obtain the information desired. These recommendations can be provided when searching for information with a search engine such as Google, interacting with an e-Commerce site like Amazon to find products and services to meet specific needs, or through interacting with other users in a social networking environment. Enterprises wish to offer products to users which can meet their needs and preferences while they interact with information systems. Users wish to efficiently utilize the information systems to find the information and knowledge

they need. So, well designed Recommender Systems can and do provide a very useful service across many areas of research and commerce.

An Interest Graph represents an area of knowledge with a graph of nodes and edges where the nodes represent the concepts and the edges represent the relationships between the concepts. For Example from medical research, the body of knowledge for the hormone Melatonin can be represented as a graph of concepts or nodes connected together with relationships or edges.

A Document Network (DN) is an information resource for communities of users who query those resources to obtain information pertaining to their interests available within the resource. These resources have multiple distinct relationships between their documents and use semantic tags or indices to classify their documents appropriately for the community of users. Some examples include bibliographic databases containing scientific publications, preprints, internal reports, as well as databases of datasets used in scientific endeavors such as the World Wide Web (WWW), Digital Libraries, or Scientific Databases (such as Medline).

The overall objective of this dissertation research is to improve computational approaches for Recommender Systems related to Document Networks using innovative solutions including the use of graph databases to represent Interest Graph for the document network. The dissertation considers broadly the requirements and available knowledge to achieve this objective. All relevant aspects are considered including the indirect associations between documents to

be searched which may not have stored associations in the DN presently. And, I consider new computational methods for performing the search and ranking of the recommendations once an interest graph has been identified for a user search request. The dissertation research selects a case study for in-depth consideration. This case study can be used to provide a methodology for finding the best solution for other such cases. An optimized Recommender System platform is developed and tested for the case while iterating the solution to improve recommender accuracy and novelty for test cases to an improved solution.

The research foundation provided supports the use of semantic technologies combined with machine learning computational techniques. It also includes those technologies available for mining graph data to build and utilize personalized semantic user models to support accurate and generalizable recommender systems. This research focuses on finding improved approaches available for mining the document network datasets to create the necessary models. It also includes the techniques to provide the best recommendation possible while trading-off real-time performance with computational costs and accuracy. User models and advanced computational methods are applied to large-scale datasets including graph databases which contain the necessary knowledge. Many of these methods build on indirect approaches for association of documents in a Document network to include the semi-metric approach through proximity graphs is a mining approach to create the necessary model.

Graph representations are particularly well suited for large dataset knowledge mining for the application to generalizable recommender systems. This research examines supervised and unsupervised graph mining methodologies to identify specific areas for improvement for recommender systems through experimentation with optional approaches which research has identified to offer the best potential for overall success. So, it has been necessary to find those techniques which can provide the best potential through research, and then define and conduct experiments to determine those which offer the best results.

Document Networks

One important narrowing of focus and scope is the type of database for the recommender system to be the subject of this project. And, as I have found in the research, conclusions and recommendations can often depend on the type of recommender system used. This research focuses on Document Networks (DN) and the relationships between documents in the networks. DN's typically function as information resources for communities of users with common interests who query them to obtain relevant information related to their activities. So, since I wanted to have the research oriented to Interest Graphs, this is a very ripe ecosystem for exploring recommender systems for users with common interests which is a big part of the original objective. To fulfill these goals, both techniques associated with Recommender Systems and Graph techniques to analyze networks of documents are needed to achieve the goals. Within the scope of

Document Networks, I focus on medical research document networks and utilize the Semantic Medline database for this research.

Graph Database for Efficient Searches

Graph databases to represent Document Networks are a more efficient way to understand the relationships represented especially for very large DN like Medline. SQL databases which provide the underlying technology for Medline really are poorly equipped to efficiently show the many relationships and to produce new document relationships quickly. Medline has millions of documents. Graph databases are often faster for associative data sets and map more directly to the structure of object-oriented applications. They can scale more naturally to large data sets as they do not typically require expensive join operations.

User Context and Interest Graphs

How does the context of the recommender query come into play around Document Networks? The context aspect for DN has to do with the different use of keywords within a given community. Each interest graph community has its own context. Indeed, each resource is tailored to a particular community of users, with a distinct history of utilization and deployment of information. For instance, the same keywords are related to different sets of documents in distinct resources, thus resulting in different distances for the same pairs of keywords.

The way documents are organized in information resources is an expression of the knowledge traded by their communities of users. Documents and keywords are only tokens of the knowledge that is ultimately expressed in

the brains of users. A knowledge context simply mirrors some of the collective knowledge relations and distinctions shared by a community of users. The distance graphs which relate elements of DN define an associative semantics. They convey how strongly associated pairs of elements in the specific network are. [62]

An information resource is characterized with sets of distance functions for the keywords common to the documents. The collection of all relevant associative distance graphs extracted from a DN is an expression of the particular knowledge it conveys to its community of users as an information resource. Notice that different information resources may share a very large set of keywords and documents. However, these are organized differently in each resource, leading to different associative semantics. Indeed, each resource is tailored to a particular community of users, with a distinct history of utilization and deployment of information. For instance, the same keywords are related to different sets of documents in distinct resources, thus resulting in different distances for the same pairs of keywords. Therefore, we refer to the relational information, or associative semantics, of each information resource as a *Knowledge Context* (Rocha 2001b). [62]

I do not mean to imply that information resources possess cognitive abilities. Rather, I note that the way documents are organized in information resources is an expression of the knowledge traded by their communities of users. Documents and keywords are only tokens of the knowledge that is

ultimately expressed in the brains of users. A knowledge context simply mirrors some of the collective knowledge relations and distinctions shared by a community of users. The distance graphs which relate elements of DN define an associative semantics to convey the strength between associated pairs of elements in the specific network. [62]

Humans use language to communicate categories of objects in the world. But such linguistic categories are notoriously context-dependent (Lakoff 1987, Rocha 1999), which makes it harder for computer programs to grasp the real interests of users. In information retrieval, systems tend to use keywords to describe the content of documents, and sets of keywords to describe the present interests of a given user at a particular time (e.g. a web search). [62]

Recommender Systems and Applications

The principal issues associated with finding information on the web have to do with the current technology utilized to address user needs. Search engines and recommender systems share a common technology foundation which is based more on what people do historically in search and with past internet transactions than it does on the actual interests and desires of users when they attempt to find information to fulfill current needs. Current needs may or may not have something to do with previous needs when visiting the internet. And, current needs depend on the current situation of the user which changes constantly during their lives often during any given day. Those current needs are very much dependent on the context of the environment in which the user makes new

requests. And, context varies for a person throughout our lives as we transition from work to non-work activities during a normal daily cycle. And, the information we desire shifts along with those transitions constantly. The past use of the internet is an inaccurate way to predict the needs of a user in real time. Surely, with the computer power possessed in today's world we can find better ways to provide recommendations for information and to enable faster search of a broader set of candidate recommendations. We all experience these frustrations with current systems today. It is not enough to just have the existing tools. But, we also need to constantly improve them to serve our needs better as we strive to improve our world for the future.

Thanks to many new internet capabilities such as Facebook and Google, we can now use social graphs which are maps to the people we know with services built around the graphs to aid our social and business interactions. But, current recommender engine approaches which only utilize the social graphs do not take into account the vast differences we have in terms of individual interests. So, a new concept has arisen to fill this gap. It is called the "interest graph" which is another kind of map which doesn't just connect us to people we know from the past but can also connect us to ideas in which we have an interest and new people who have similar interests. A combination of social and interest graphs is the new concept. Some social networks attempt even now to provide capabilities in the interest area in a variety of ways to make these interest connections. But, so far at least, the inclusion of "interest graphs" is just in the infancy and has not

fully made use of the technology of the semantic web concepts which are still to roll out from academia and from enterprises which have invested in these new approaches. These new approaches are clearly coming as the Semantic Web becomes realized over time and as new capabilities are introduced to the internet by enterprises with the vision and resources to make that happen.

Knowledge Context is an approach to combine the interests of users and the social context of the terminology around the interests. This approach actually gives us a very good combined approach to pursue the interest graph and context together. This Dissertation will focus on the interest graph side of Knowledge Context.

Enterprises are developing many new ways to build an interest graph. The most obvious is to simply ask people about their likes and dislikes and build that into a profile for the individual user. Another more difficult way is to infer those likes and dislikes based on past interactions. This takes more computational power and sophisticated algorithms to achieve. For example, if I search on a term like say Thomas Jefferson, an inference capability might record that I have a strong interest in Thomas Jefferson especially if I did this multiple times. But, I may only do it a few times to complete some academic report and not actually have a real long term interest in Thomas Jefferson. But, the inference would be recorded anyway and be part of my individual user profile going forward when it is not a long term interest. So, inference has its limitations clearly as well. All of our commenting, liking, searching and foraging for information on the web

provide our own unique trail which can be utilized for making future recommendations and future searches. But, what is the best way to utilize this information for accurate use in the future. This is the holy grail now being researched by many companies and it leads to further disruption in the future of the internet as new techniques become discovered and are implemented going forward. How do we capture our meaning in a way which is useful and has generalized capability for future web interactions?

For a new approach like an “interest graph” to emerge, I am referring to concepts which represent people and things as opposed to actual people and things. It is this distinction which connects our interests to the new semantic web which is the next big evolution of the web which is moving towards infusing meaning into text and other objects as a method to easily automate connections to ideas. There are many ways in which these connections to concepts are formed including past searches and transactions on the web. Historical trails are one way to infer our interest but it only takes us so far in the quest. For example, Google tracks past semantic connections now with that it calls the “knowledge graph” and sometimes surfaces them in our search results now. It is essentially based now on our past searches to connect ideas and it does build an individual user knowledge graph which is a great strategy because it is very difficult to replicate today. Google does have a competitive advantage here now. Soon, Google will be able to connect these past search concepts to discover new interests we didn’t realize we had ourselves. This is possible with the semantic

connections included in the Google Knowledge Graph. For example, a past search for Thomas Jefferson could also include a recommendation to consider Thomas Paine who has a semantic connection to Thomas Jefferson when the user has never searched for Thomas Paine in the past. This dissertation will expand on these new indirect connections and their inclusion in Recommender Systems.

From a more commercial perspective, if a search engine might know that when I search for a nearby park, one of the activities generally enjoyed in parks is to have picnics. Given that kind of automated understanding, the search engine might present information on delis and bakeries with tempting advertisements for me to buy my picnic supplies. This is not actually a revolutionary marketing approach. But, it is marketing which could be accomplished automatically by utilizing creative insights from the semantic web whereas today it is more human generated. And, it is a capability which can be utilized by more companies in the future as recommendations are made and search results are presented. This all gets more interesting as we connect these possibilities to the new concept of an “interest graph” to support our interactions on the internet.

Today there are nearly half a trillion publicly available social network relationships of people and things in what could now be considered our presently available “interest graph”. And, it grows by more than 2 billion points each day as it has been estimated by Facebook. We are current sitting already on a wealth of

data which is being utilized for recommendations and search. There are several ways in which this data can be best used.

Ad Matching

Analysis of the currently available “interest graph” allows companies now to deliver content which may be more relevant to you based on publicly expressed interest. And, just this can achieve much better ad matching with the correct consumers by a factor of 50 to 100 times better than previous methods.

E-commerce Recommendations

Just as advertisers segment audiences now, e-commerce sites can group users into categories for recommendations based on people similar to you. This can help when the site doesn’t have past information on a user specifically but still wants to recommend products and services to get best response. These sites add interest information to existing algorithms in the form of an interest signal derived from someone like you based on social network data hoping to provide future recommendations which would be of interest even though they may have little past data on the specific user.

Customer Relationship Management

The Interest Graph can deepen relationships with loyal customers by helping to find information as to those users actually interests and using that information to deepen the bond. The interest graph can provide the additional information which may not be already available to help deepen the loyalty which all translates to more future business.

Entertainment Applications

Most people don't have the time to patiently organize movies just to be given suggestions for new movies they may like. Entertainment applications too, can benefit from the wealth of Interest Graph data for understanding what people like and improving recommendation engines.

Marketing and Promotions

Traditional marketing and promotions are built on segmentation of the market where assumptions are made about consumers based on demographics and group behavior. But, only a certain level of accuracy can be obtained with segmentation approaches because interests are much more diverse than those captured by typical segmentation approaches. Knowing someone's actual interest can greatly improve the way promotions are executed to make marketing budgets go further and customers happier. We are only beginning to understand the profound impact of the "interest graph" and it's potential for creating and deepening relationships.

Some stretch the term "interest graph" to include other people who share our interests say within a social graph. Facebook actually does this now with the idea of Like. But, the interest concept is actually a substantial issue on its own and current capabilities within Facebook just scratch the surface of the "interest graph" presently. This figure gives a better idea of how social and interest graphs need to be combined in the future. **[80]**

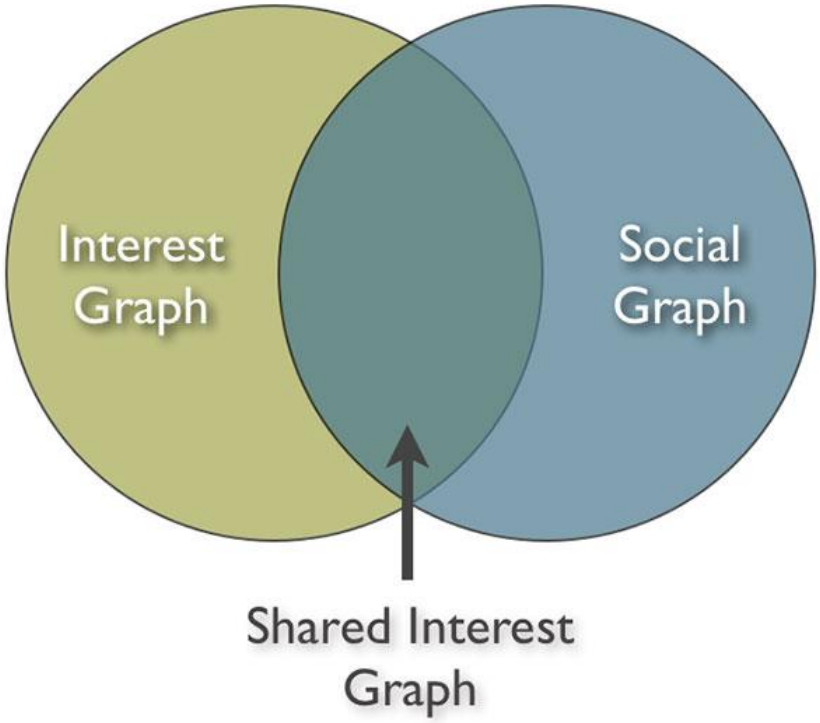


Figure 1 - Shared Interest Graph [80]

The combination with the social graph needs to be at the most basic level. It needs to be a shared interest graph in two ways: 1) finding new *interests;* and 2) finding new *people* or a map of people who share your interests. **[80]**

For finding new interests, the shared interest graph has possibilities. The Facebook Graph Search doesn't really help with new interests but it can help to find people with new interests once you define it on your own. It can combine finding new people and finding new interests. But, you do it by visiting profiles of

people who share an interest for something else and see what else they are interested in. This is similar to the concept of Collaborative recommendations. Research has shown it does get some good recommendations but not as good as can be obtained when utilize semantic approaches as well. So, there are better methods and this research is all about finding them, experimenting with a few, and recommending the best approaches to incorporation of semantic methods with recommender systems for the inclusion of context. It is a matter of accuracy and we hope to find the best approaches possible with the least work involved both from the perspective of the computational model and with respect to work by the users. So, inference does need to be included as much as possible to strike a good balance. There are no ideal approaches though and compromises need to be made for sure. **[80]**

None of this is actually new as companies like Amazon have been trying to find good approaches for recommender systems for years now. Amazon has had very detailed schemas for describing products which have built over the years. These early approaches have been in the direction of the interest graph but have all been based on proprietary standards. The benefit of the semantic web is it then is based on publicly available and non-proprietary standards so no one company would have a competitive edge once the ontology is defined. This is essential in order to get massive buy-in to the concept and to get users to spend time building the ontologies required. Companies are beginning to standardize semantic descriptions even though they are fierce competitors. This

can force companies like Amazon to open up their product databases to future semantic search engines or to offer such semantic search engines themselves.

This means that the “shared interest graph” could become more portable between all the major services like Google, Facebook, Amazon, and future major internet companies providing search and recommendations. History shows that commoditization does happen over time and with the semantic web inclusion, this could become a very powerful force in the future. But, we still come back to how this can best be accomplished. And, so far, there are no final answers here. There is a lot of research on various approaches. No specific approach has been universally adopted because this is a very complex undertaking with the need for an all-inclusive user model required at the core of the technology which is populated largely through inference. This is not a simple undertaking. It will take years to transpire with numerous research projects to advance the technology still lying in the future. **[80]**

The technology of the shared interest graph has grown to a large extent out of finding new products and services to buy. This is usually how a new advance starts since it is where the money is and such software developments efforts do need to be funded in order to be launched. But, once this infrastructure is built for products and services, then it helps us to be much more than just better consumers. It has the potential to help us connect to new interests and new people who share those new interests in very profound ways to build something much bigger eventually in the very nature of the way we interact with

knowledge for the future. So, the future is bright for those who can help to find the best approaches and who can help to transform the current internet capabilities into an even more useful capability for information going forward as the semantic web becomes more embedded in our daily lives.

One new example is from a startup called Pearl-trees. They have received investor funding for their concept based on the Peal Tree service. A Pearl is basically a bookmark where users can assemble these pearls into trees based around a topic. Pearl-trees use that data to determine how different topics and bookmarks are related and gives users the ability to find new pearls through a button called “related interests”. They name this new capability “tree rank” after Google’s “page rank” and Facebook’s “edge rank” technologies. So, in essence, it is offering an “interest graph” for general access on the web. So far, it has grown very quickly in this area from 2009 to today with millions of Pearls available now.

The point is that companies are experimenting with a variety of approaches but none is clearly being adopted on a wide-spread basis presently. What is the best approach to using a shared interest graph to improve recommender systems and to incorporate context into the process with semantic approaches?

This Dissertation focus on the “interest graph” side of this intersection as it relates to Document Networks. The inclusion of the “social graph” side will be left to other research. But, this Dissertation does provide important advances in this

focus area which can later be included in another project which also includes the social graph. This research includes these resource elements to provide capabilities required to adequately explore this topic:

1. Item content and user content modeling with semantic technology,
2. Graph Database computational techniques for search and visualization,
3. Mathematical approaches for creation of algorithms to discover new and important relationships between the documents in a network,
4. Experimentation with alternative approaches to combine important recommender methods to achieve results which improve the user recommendation experience.

Research Requirements and Gaps

Graph Databases are just beginning to be used for Document Networks (DN) although there is great interest. The use of Graph databases for applications in general is relatively new. Available DN search tools today have a number of deficiencies using their current technology bases. Some of these gaps include the following:

- Can miss indirectly related documents with strong associations.
- Are not based on the proximity or similarity measure to rank associations with user needs.
- Are not predictive to make use of machine learning tools to train a model and use the model to predict associations.

- Do not make use of individual user characteristics and their impact on the results.
- Do not utilize the knowledge base of Recommender Systems and graph database tools.

Also, other gaps include the fact there are very few comprehensive studies published which use graph databases and Recommender Systems tools for DN citation recommendations. There needs to be considerably more research in these areas to make sure the vast knowledge available today through these resources can be efficiently and accurately used with better technology. There are many new discoveries which can be made with the information we already have but which is difficult to use because of the lack of research currently. Also, the are many trade-offs associated with migration of a large DN database with millions of citations to a graph environment, and the methodology for applying recommender systems and graph databases to DN which has not been well established since these are relatively new technologies. So, this dissertation research narrows or eliminates these many gaps in the research and provides a comprehensive approach and methodology for transforming existing DN resources to help provide greatly improved recommendations in the future.

Research Hypothesis and Dissertation

The **research hypothesis** is that computational approaches for Recommender Systems will improve the accuracy and novelty for document networks through the use of (1) graph databases to represent the document

network, (2) advanced computational algorithms for graph databases, (3) novel indirect relationships between the concepts in the graph DN, (4) machine learning tools to predict links and ranking of results, and (5) user information such as context and interests.

Dissertation – The use of Recommender Systems technology combined with Graph Database technology when applied to Document Networks to improve the accuracy of predicting the 'rating' or 'preference' that a user would give to an item of information from the document network over conventional approaches. And, it also finds new important recommendations not presently found without these technologies.

I began this dissertation project with a broad approach in mind knowing it would need to be narrowed as the research proceeded. The broad approach had several types of large databases and technical implementation approaches in mind. It also foresaw reaching into quite a number of concepts associated with recommender systems which is a very broad research topic. After considerable work preparing the technical environments and drilling into the research concepts and databases, I have narrowed the focus to a few specific areas for research which could be contained in a single dissertation research project. This need to narrow the topic was well understood from the beginning. So, I have now narrowed the focus as the research transpired while preserving the original goals of the research into areas related to recommender systems and specifically improving the accuracy with better approaches. These improved approaches

draw on the same body of knowledge proposed originally although for the purpose of this Dissertation I am more oriented to the computational methods and oriented to a specific type of Recommender System since the type determines the best methods.

Problem Outline, Approach, and Questions

This section provides an overview of the research conducted for this research proposal. It first describes the background and motivation for the research, followed by the problem outline for the thesis. Next, it provides a brief description of the research context of this thesis, and states its research questions. This is followed by a brief description of the research approach and the research contributions. Finally, the papers included for the proposed research are listed, and a brief overview of the structure of the rest of this proposed research is given.

There are several elements which must be explored to offer solutions to the problem of context aware recommender systems which achieve accuracy in recommendations for generalized content. Some of the key research areas are as follows:

1. Semantic models which can be populated through inference and which include the context for the user which is valid at time of information request.
2. Methods which can efficiently mine large datasets to build semantic user models.

3. Methods to efficiently mine large datasets to build semantic models for knowledge which may be the items recommended to users.
4. Query approaches which can efficiently search the personalized user model and knowledge to accurately provide recommendations through these systems.

The combination of these elements provides the basis for inclusion of context in recommender systems while also improving the accuracy of recommendations themselves. Clearly, semantic techniques offer the best overall solution to these problems. But, semantic approaches are also very challenging in terms of building the ontologies required to produce the recommendations efficiently in real time to answer questions posed.

The research first reviews the current research in these areas to find the best methods already discovered in the literature. Then, new approaches to these problems are prepared and researched in order to have the best set of optional approaches possible for solving the overall problem best. And, an experimental approach is prepared to test each of the best optional approaches to discover the advantages and disadvantages of each. Finally, the research explores the best approach discovered to move that approach to a proof of concept stage to conclude with a recommendation for implementation of the best approach possible.

I know from the research to date that the use of semantic techniques is very beneficial. I also know that there are two basic semantic approaches as the

formal approach using ontologies and the more informal approach using Folksonomies.

There has been research into these different approaches and combining them to find a more efficient method which is to some extent a compromise for achieving a system with good performance while balancing the cost and benefits. So, this research brings the options chosen for experimentation to a very meaningful combined approach which optimizes the recommender system.

Past research brings me to the options which are most worthwhile to explore in-depth. And, the in-depth experimentation is conducted with care using good scientific method approaches and controls so the results and conclusions are well supported by the experimental approach conducted. I make use of existing resources available to GMU for testing these approaches. Those resources have already been cited previously.

Research belongs to one of three possible approaches; exploratory, testing-out, and problem-solving. Exploratory approaches are concerned with breaking new ground in terms of studying new problems/issues/topics about which there is little knowledge. This is not the case here since there is already a great deal of research in these areas over a long period of time both in the academic and corporate world. Testing-out research, on the other hand, studies already existing generalizations and tries to find limits or new application areas for the existing theories. This research is not just about testing-out existing theories to understand them better. It is about finding new innovative solutions to

these real world practical recommendation problems. This research is the latter type of problem solving research, which takes as a starting point a real world problem, defines it properly, and finds a solution to it by applying a range of methods. The research proposed here falls into the latter category, problem-solving, since our starting point is the problem of generating a framework for the inclusion of context in recommender systems that combines both formal and informal semantics with machine learning using large graph datasets.

The research is based on analyzing a problem, designing and implementing a solution to it, and evaluating the problem solution by an experiment. In some sense the research has been iterative, since the results from each analysis, design, implementation, and evaluation cycle have been fed into the next phase of the research to provide a new starting point with new knowledge.

The research consists of several main phases. The phases explore the 4 main research areas listed above and be based on previous research in to each of these areas to find a good starting point and to define optional approaches for consideration also yet to be defined. The initial phases of the research are oriented to assess and validate best optional approaches to the overall problem and the four main areas which comprise the overall problem. These approaches are evaluated for best single approach which could produce the best results when combined together in a fully operational recommender systems approach. And, the key optional approaches are the subject of experimentation to

determine the further explore and define the costs and benefits of the proposed solutions. Finally, a single best approach is found and demonstrated with a proof of concept pilot system to substantiate the recommendation capability.

This research answers many questions associated with this research topic as follows:

1. What are the best alternative computational models to represent a user's interest graph in a given subject? The research has several computational models referenced. Which of these offer the best starting points to identify the best optional models for the research? If none of these provide a good starting point, what would be a good starting point for providing such a model?

2. For these optional computational models, how can they best be implemented through inference or other means?

3. For these models, how can they be used to support recommendations? What are the best techniques for this?

4. What experiments can be designed to offer conclusive experimental results to support the best candidate methods for utilizing context in recommender systems with these semantic user models?

These are the broader questions. I also provide questions to a more detailed level after the background research is presented and the optional approaches for experimental design have been defined. Then, this dissertation will pursue those approaches to obtain an optimized algorithmic solution.

CHAPTER TWO

Methodology

The dissertation follows a methodology which explores best practices and iterates to result which fills gaps and meets requirements. The methodology itself provides a roadmap for such Recommender Systems to use for new Document Networks. And, it is based on a case study for Semantic Medline with is a medical research Document Network serving medical researches.

The steps I follow for the methodology for the Case Study is as follows:

- **Migrate** Semantic Medline database from its current MySQL database to Neo4J, a leading graph database, using the MySQL backup files.
- **Restore MySQL backups**, build MySQL queries to prepare Neo4J import, execute Neo4J import, and validate import results.
- **Research and select methods to interface with Neo4J** to implement Semantic Medline search algorithms with the new graph database.

- **Research and select the best optional search algorithms** to improve accuracy of the recommendation results once in a graph database.
- **Develop and implement a user interface** with embedded search algorithms using the optional algorithms selected.
- **Prepare test cases** for experimentation to achieve best recommendation results.
- **Validate the test cases** to original the Semantic Medline MySQL database to make sure platform has a proven result.
- **Prepare and conduct performance testing** to show relative improvement for each option. The performance tests results are compared to a baseline reference standard validated by experts in the test cases.
- **Adjust, and iterate solution** to improve performance results as measured with the performance tests.
- **Provide an analysis** of the case study within the broader context of the dissertation to explore overall conclusions.

These steps in the methodology are discussed further going forward.

Semantic Medline Document Network Case Study

Semantic MEDLINE is a Document Network web application hosted by the National Library of Medicine. It summarizes MEDLINE citations using a Natural language processing front end to extract semantic predications from MEDLINE

titles and abstracts. The predications are presented in a graph visualization tool that has links to the MEDLINE text processed. The database is stored in SQL tables with the table structure shown on next.

MEDLINE consists of over 20 million medical research publications, which is a BIG DATA document database! The Semantic Medline database is a brilliant compact representation of the linked knowledge concepts that are contained within that huge document network. Semantic Medline goes far beyond and deeper than traditional citation networks (which contain only the authors' names and manuscript title and high-level keywords). For example, a Predication example is “BRCA1 Associated_With Sporadic Carcinoma”.

Semantic Medline is pre-processed to extract and summarize semantic predications (RDF triples in format ‘subject-predicate-object’ triples) from the text of the documents in the DN. It was developed for biomedical research literature using domain knowledge provided by the Unified Medical Language System (UMLS). It represents textual content with the UMLS concepts as arguments and UMLS Semantic Network relations as predicates. Semantic predications represent the source text where the SemRep predications from multiple documents provide input to the Semantic Medline summarizer to provide the reduced and focused list of predications also called “semantic condensate”. Each semantic condensate is based on a user selected topic and a summarization perspective. Each perspective is represented by a set of formal constraints on the arguments and on the predicate of the input predication. The transformation

from the initial query list of predications to the reduced list of semantic condensates is guided by Relevance, Connectivity, Novelty, and Saliency. Therefore, search solutions already have a core semantic environment. Medline has recently expressed interest in the use of a graph database like Neo4J for implementation along with a comparison to current MySQL tools.

Semantic Medline is a system which is based on relationships between documents discovered with automated tools and enables knowledge discovery through interactive visual maps of linked concepts among medical research documents. The core of Semantic Medline are two important tools; SemRep, which extracts semantic predications (subject-predicate-object triples) from text and an automatic summarizer. SemRep predications comprise executable knowledge which can be stored as a graph database and are amenable to automatic manipulation with the graph database. [61] The automatic summarization provides a reduced and focused list of Semantic Condensates based on a user-specified topic and summarization perspective represented as a set of formal constraints on the arguments and the predicate of the input predications. The summarization from initial list to the reduced list of Semantic Condensates is guided by Relevance (conform to the selected summarization perspective), Connectivity (include predications which share arguments to the selected summarization perspective), Novelty (eliminates known predications), and Saliency (eliminates predications with low frequency of occurrence).

The Semantic Medline user interface has 4 tables for Search, SemRep, Summarization, and Visualization. The Search tab gives the user a way to extract records for a concept from the Medline database. The SemRep tab presents the predications extracted in the Search tab. The user can then go to the Summarization tab to select a topic and perspective which further refines the search in Semantic Medline. The Visualization tab uses a graph tool to represent the summarized semantic condensate and which guides the navigation through the actual documents retrieved for the search. Nodes and edges are color-coded based on UMLS groups. [65]

The relationship between documents in a DN and keywords allows users of the information to infer the semantic value of documents and the inter-relationship of the document keywords. In an academic DN like Semantic Medline, these relationships refer to document citations and the keywords refer to subject-predicate-object triples of the SemRep semantic predications. The closeness between keywords and the documents they classify can be computed as the Keyword Semantic Proximity (KSP).

Medline Database Attributes and Discussion

This Dissertation concerns a few key research areas where the background needs to be established clearly to provide a foundation for the research focus. The research is focused on Document Networks using the Semantic Medline database for experimentation. These research areas consist of the following:

1. **Semantic Medline Document Network Case Study -** What is it and what are the characteristics which make it unique as a Document Network? How do these traits impact the study here?
2. **Context of Medline Document Network as a Recommender System -** What are they and which type am I researching in this study? What are the characteristics of the type I am studying here?
3. **Graph Databases and Indirect Associations improve Value of Semantic Searches -** What do they provide which is different than previous forms of data storage and how these characteristics be important to the study here? What techniques are best used from Graph database analysis to support the goals of the study?
4. **Evaluation approach for Medline Case Study -** How are Recommender Systems evaluated and which methods are applicable in this study?

The entity-relationship diagram for Semantic Medline is presented below. Because the files are very large and time consuming to manipulate, I did make good use of the Predication Aggregate table for the experiments developed to exercise the database. I explain in detail how this experimental design was structured in a later section. But, it is very important to realize what is included in Semantic Medline and how these tables are constructed in order to understand the approach for this Dissertation which utilizes this database as the case study Document Network.

Name: **PREDICATION_AGGREGATE** table
This table is a convenience table that joins the salient information from all the above tables for efficient access.

The entity-relationship diagram of SemMedDB is shown below graphically:

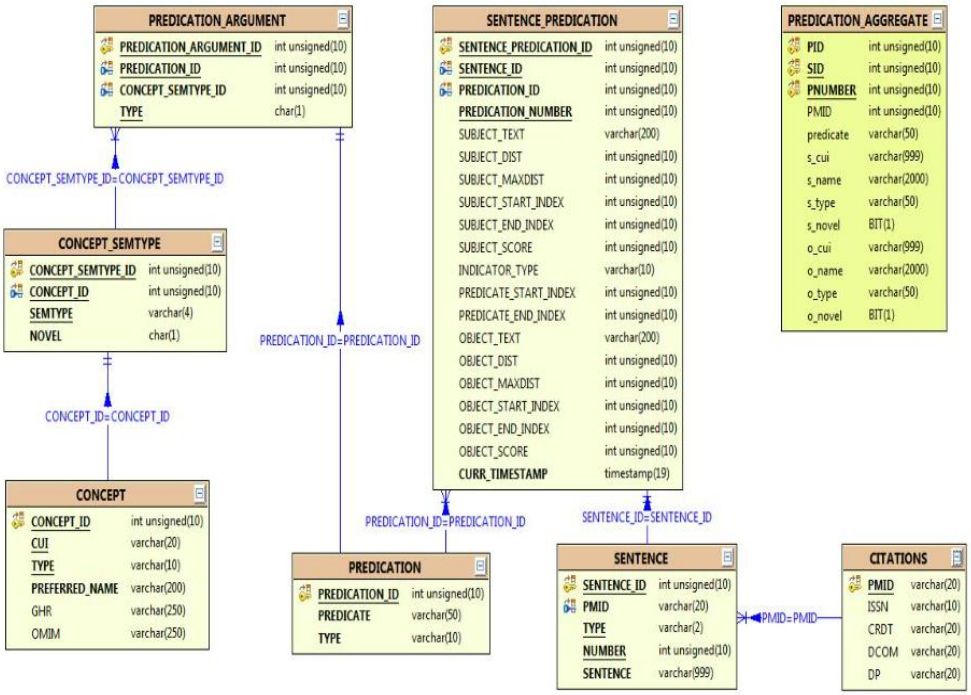


Figure 2 - Semantic Medline Tables [61]

Tables:

Name: **CITATIONS** table
This table contains relevant metadata for each PubMed citation and has the following data fields:

PMID: PubMed identifier of the citation

ISSN: ISSN identifier of the journal or the proceedings where the article was published

DA: Creation date for the citation

DCOM: Completion date for the citation

DP: Publication date for the citation

PMID	ISSN	DA	DCOM	DP
19851774	1432-203X	2010 01 21	2010 03 18	2009 Dec

Name: **CONCEPT** table
This table contains information about the UMLS Metathesaurus concepts as well as EntrezGene terms used by SemRep. In the current version, UMLS Metathesaurus concepts are from the UMLS 2006AA release. Data fields in this table are as follows:

CONCEPT_ID: Auto generated primary key for each concept

CUI: Concept identifier (CUI) of the concept, corresponds to UMLS CUI if it is from UMLS, and the gene identifier from EntrezGene if it is from EntrezGene

TYPE: "META" if it is a UMLS Metathesaurus concept, "ENTREZ" if it is an EntrezGene symbol

PREFERRED_NAME: UMLS Metathesaurus preferred name for the concept, or the official gene name from EntrezGene

GHR: Corresponding Genetics Home Reference (GHR) identifier, if the concept is a gene or a disorder

OMIM: Corresponding Online Mendelian Inheritance in Men (OMIM) identifier, if the concept is a gene or a disorder

CONCEPT_ID	CUI	TYPE	PREFERRED_NAME	GHR	OMIM
1844	C0003873	META	Rheumatoid Arthritis	NULL	180300:604302
1276072	215	ENTREZ	ABCD1	NULL	NULL

Figure 3 – Semantic Medline Table Definition 1 [61]

Name: **CONCEPT_SEMTYPE** table
This table links concepts in the CONCEPT table with their semantic types. A concept may have multiple semantic types. There is a 1-to-many relation between the CONCEPT and CONCEPT_SEMTYPE tables. The data fields are as follows:

CONCEPT_SEMTYPE_ID: Auto-generated primary key for each concept-semantic type pair

CONCEPT_ID: Foreign key to the CONCEPT table

SEMTYPE: UMLS semantic type abbreviation, such as aapp (Amino Acid, Protein, or Peptide) or gngm (Gene or Genome). For the list of all abbreviations, see SRDEF.

NOVEL: Identifies whether the concept is novel or not. Novelty of a concept-semantic type pair is computed based on its distance from root of the UMLS Metathesaurus hierarchy and has been used in automatic summarization approaches based on SemRep [1].

CONCEPT_SEMTYPE_ID	CONCEPT_ID	SEMTYPE	NOVEL
2628	1844	dsyn	Y
1481123	1276072	gngm	Y

Name: **PREDICATION** table
Each record in this table identifies a unique predication. The data fields are as follows:

PREDICATION_ID: Auto-generated primary key for each unique predication

PREDICATE: The string representation of each predicate (for example TREATS, PROCESS_OF)

TYPE: Can be ignored

PREDICATION_ID	PREDICATE	TYPE
87120	PROCESS_OF	semrep

Figure 4 – Semantic Medline Table Definition 2 [61]

Name: **PREDICATION_ARGUMENT** table
Each record in this table links a unique predication with one of its arguments. There is a 1-to-many relation between the PREDICATION and PREDICATION_ARGUMENT tables. The data fields are as follows:

PREDICATION_ARGUMENT_ID: Auto-generated primary key for each predication argument

PREDICATION_ID: Foreign key to the PREDICATION table

CONCEPT_SEMTYPE_ID: Foreign key to the CONCEPT_SEMTYPE table

TYPE: 'S' for subject argument and 'O' for object argument

PREDICATION_ARGUMENT_ID	PREDICATION_ID	CONCEPT_SEMTYPE_ID	TYPE
176604	87120	2628	S
176605	87120	21437	O

Name: **SENTENCE** table
This table contains information about individual sentences from PubMed citations and includes the following data fields:

SENTENCE_ID: Auto-generated primary key for each sentence

PMID: The PubMed identifier of the citation that the sentence belongs to

TYPE: 'ti' for the title of citation and 'ab' for the abstract

NUMBER: The location of the sentence within the title or the abstract

SENTENCE: The actual string of this sentence

SENTENCE_ID	PMID	TYPE	NUMBER	SENTENCE
113049226	19855969	ti	1	Rheumatoid arthritis in patient with homozygous haemoglobin C disease.

Figure 5 – Semantic Medline Table Definition 3 [61]

Name: **SENTENCE_PREDICATION** table
This table links a sentence with the predications extracted from it. There is a 1-to-many relation between the SENTENCE and SENTENCE_PREDICATION tables. It includes the following data fields:

SENTENCE_PREDICATION_ID: Auto-generated primary key for each sentence-predication pair

SENTENCE_ID: Foreign key to the SENTENCE table

PREDICATION_ID: Foreign key to the PREDICATION table

PREDICATION_NUMBER: The number of times the predication is extracted from the sentence. If there are two instances of the same unique predication in a sentence, the value is 2.

CURR_TIMESTAMP: The timestamp for the record

The rest of the fields in SENTENCE_PREDICATION table provide mention-level information for the elements of the predication (predicate, subject, and object).

INDICATOR_TYPE: The type of the predicate, such as VERB for verbal predicates, and NOM for nominalizations and other argument-taking nouns. For a full list of indicator types, see the Appendix in [2]

PREDICATE_START_INDEX: The first character position of the predicate mention

PREDICATE_END_INDEX: The last character position of the predicate mention

SUBJECT_TEXT: The subject mention in the sentence

SUBJECT_DIST: The distance of the subject mention (counted in noun phrases) from the predicate mention (0 for certain indicator types, such as NOM)

SUBJECT_MAXDIST: The number of potential arguments (in noun phrases) from the predicate mention in the direction of the subject mention (0 for certain indicator types, such as NOM)

SUBJECT_START_INDEX: First character position of the subject mention in the sentence

SUBJECT_END_INDEX: Last character position of the subject mention in the sentence

SUBJECT_SCORE: The confidence score of the mapping between the subject mention and the subject concept

OBJECT_*: The fields representing information about the object, in the same way the SUBJECT_* fields do for the subject

SENTENCE_PREDICATION_ID	SENTENCE_ID	PREDICATION_ID	PREDICATION_NUMBER	...	CURR_TIMESTAMP
57109318	113049226	87120	1	...	2011-11-17 19:58:38.0

Figure 6 – Semantic Medline Table Definition 4 [61]

Steps to Improve Semantic Medline Recommendations

Selecting Semantic Medline as the database for performing a DN case study:

1. I **researched and experimented with alternative computational models** available from Recommender Systems and Graph Database technology to apply to Semantic Medline.
2. From these experiments and research, I found the **best starting point** to begin a solutions oriented development process to iterate the best optimized models for Semantic Medline.
3. If the computational models did not provide improved approaches, then I **found other models** which could be more promising.
4. For the best computational models discovered, I then found the **best way to implement them** for Document Networks like Semantic Medline.
5. I then designed an **experimental process** to provide conclusive proof as to the best approaches available to achieve the dissertation objectives.

Once implemented and tested, I continued to iterate to achieve the best results feasible for the optimized DN Recommender System algorithm

Migration from MySQL to Neo4J

So with this research focus, I actually imported all this data to Neo4J from .sql backup files. First, I have to load it into MySQL from the backups. That alone took weeks to accomplish with these huge files because of the processing required to import the backups and the extent of the millions of records included.

Once in MySQL, I could then output .csv files I wanted for import to Neo4J. This took a long time too with many back and forth sessions to tailor the exports and to import it properly in Neo4J using the structure there. And, both of these systems were not well known to us when starting. However, over the course of many trials, I did get to point where it became well known and I could function in these new environments. So, it is important to point out that just the import of Neo4J was a huge undertaking from a .sql backup file with millions of records. Here is our MySQL environment:

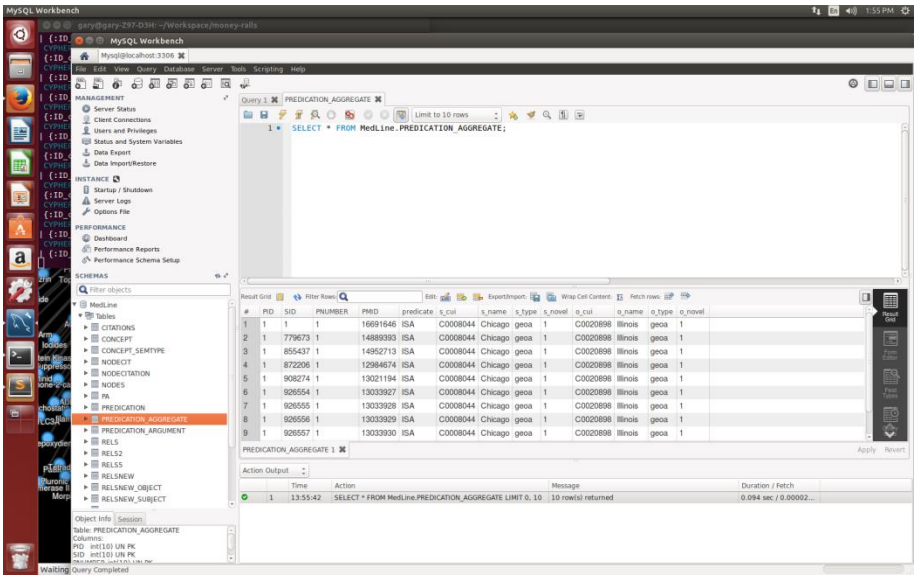


Figure 7 - MySQL Data Migration

I spent many weeks trying to get this migration accomplished!! It is very slow with millions of records!! The Citations to the actual documents are included in the PMID which is a field in the Sentence table linked to the PMID in the Citations table. The Sentence Predication table includes the links to the Sentence table through the SID for Sentence ID and combines it with the predications and concepts to get the triplet or RDF structure of Semantic Medline. The Predication Aggregate table is a summarization of these connections and includes the key data for the RDF structure.

In a graph database, the concepts become the nodes, the predications become the edges, and they tie to citations which are sentences in the abstract of the actual documents. The Sentence predications themselves are compiled using the SemRep modeling approach to summarize the sentences with concept-predication-object. This captures the essence of the relationships including the semantic and context aspects for the millions of predications which result from the SemRep process. The Predication Aggregate already has many tables joined to provide the resulting RDF table without having to run the joins involved to combine all the tables using SQL. I ran joins some of the tables to make our own aggregate tables and found the joins with the millions of records take days to complete on a typical system. So, it is very good to have the Predication Aggregate already available with the joins completed to save computational time. And, then use these to compute association distances using the optional approaches discussed in the next sections.

Since the Predication Aggregate includes the PMID, I work mostly with this table in our research which has the direct tie to the sentence and to the actual citation on a one-to-one basis. Inclusion of all the sentences and citations makes the Neo4J database quite large and slow for queries. So, I am dropping those here since I am working with a technical infra-structure not capable of their inclusion and still have timely execution of processes. I did experiment with their inclusion but found it to be basically unwieldy. So, it was necessary to drop them for the experimental framework. And, this does not change any of the results since it is just a simple lookup to find these citations from the PMID which is stored with the records in the Predication Aggregate table anyway.

From the perspective of Relevance, Connectivity, Novelty, and Saliency, how can proximity and distance best be utilized to improve the quality of the findings for a given query? These evaluation criteria are clearly defined in the Appendix under the Evaluation Criteria sections. For this, I need to know a lot more about the user stories and use cases for Semantic Medline. I discuss these more in the next section here.

Context of Medline Document Network as a Recommender

For Semantic Medline, much of the context information is built into the way in which the Semantic Medline database is constructed with the RDF triple predication which combines concepts with predicates and objects using the MESH medical language structure. Medical researchers are the principal users for Semantic Medline. They represent a broad class of medical specialties from

all aspects of healthcare delivery and from the scientific community working to find new disease relationships to help create improved diagnostic and treatment approaches. The Predication Aggregate table available from Semantic Medline includes a contextual breakdown of the sentences included in the various citations. So, much of the work has been already accomplished by how it is created. The tools selected to explore the knowledge contained in Semantic Medline gives us the best opportunity to improve the matching of user context to medical citation. And, there are a wide number of tools available to do this with computational and machine learning techniques using the graph database which represents the Predication Aggregate information.

Referring to the background research on Context in Recommender Systems found in the Appendix, we see a number of areas which need to be included in order to address the issue of context. Some of the principal areas discussed are Individuality, Time, Location, Activity, and Relations. Of these, for a typical medical research user, time and location are going to be much less significant. The Individuality of the interest, the activity in which the research is directed, and the relationships of the information to other information are some of the more important context related aspects to consider. These actually do get picked up very well with the RDF approach provided by Semantic Medline which includes the predicate in the Predication Aggregate representation. The Predication Aggregate does represent sentences to the concept – predicate – object level. And, this level clearly picks up the aspects of the activity and the

individuality of the search being conducted. The relations are something I can provide by associating the concepts and predications with others which may have important connections by how I create the recommendation results and rank them to provide as output to the user. And, providing clustered results to user searches which put associated information into common clusters and providing the members of the shared cluster to the user as recommended areas to consider in terms of the user context.

Also found in the Appendix on Context, there is a discussion regarding Context transition and steps in establishing context relationships. These concepts too are actually embedded in the Semantic Medline approach through the use of predications which are employed to associated subjects and objects. So, the predications can change for connected concepts and the relationships can be established based on how the concepts are related to each other. Again, because of the way in which the predication aggregates are established, much of the capability for these context considerations are available to utilize with the Semantic Medline database approach. But, they are available the from perspective of a medical researcher without the kind of context variability which may be the case with the general public consuming web pages for example. So, within this user group, the Semantic Medline approach to modeling the citations covers many of the areas for context consideration found in the Appendix.

The graph database is the repository here for the Predication Aggregate information where concepts and predications can be networked to find interesting

connections. With this, we can utilize measures of proximity between the concepts and predications to recommend results which would not be directly available otherwise. See below the network nature of the user models which have concepts and edges associated with them. Graph connections between the concepts and predications ties directly into this kind of network model of the user. It provides results which match up to the user interests and help the user to define their interests by shifting through associations not readily apparent from straight query searches which don't include associations into the search results. With graphs, this is easily provided to match user needs and give the user a chance to select relationships which they could not accomplish with previous flat file kinds of search capabilities. So, the use of graphs is an important tool to immediately provide a good matching technique for user context as it relates to item content with medical research databases like Semantic Medline. Then, the structure of the system model can be based on these components using graphs.

Algorithm Development

Thorough research indicated the best algorithm potential which met the requirements include the following:

- Semi-Metric algorithms which include directly connected nodes and indirectly connected nodes to show new associations which may be novel for the research queries being conducted.
- Distance measures built on proximity calculations to provide a measure of predication association. Computed distance provided with the item.
- Item-based algorithms which seek to list the predications which result from the graph query in a table which can be exported and further analyzed.
- Rank algorithms which iterate a value for the predication rank using both node and edge formulas.
- Create a hybrid algorithmic solution which blends the advantages of the component algorithms into an integrated solution.

Two development environments were selected to implement the algorithms which include:

- Python using the PyCharm IDE for platform flexibility to optimize solution
- Ruby on Rails for creation of a web enabled platform with graph visualization.

These two platforms are developed to implement and iterate the solution based on results and performance testing for eventual scaling to production web implementation.

Graph Databases and Indirect Associations

Graph Database Tools – Neo4J for Visualizations

The key Recommender Systems technology powering real time recommendations is the graph database. Graph databases outperform SQL and other NoSQL technologies for connecting entities for a wide variety of purposes. Here we are focused on connecting documents in a Document Network where we can use the graph database to capture associations both direct and indirect for ranking on a search presentation. Recommender systems which use graph databases are thousands of times faster with a fraction of the coding required to implement the recommender engine than with previous approaches. Medline is implemented with MySQL which is very slow by comparison. Potentially MySQL can be used to get comparable results using JOIN constructs. But, with millions of records, such JOIN commands would run extremely slow if they run at all.

In order to build a faster engine for experimentation, I first wanted to move Semantic Medline to a graph database. The one which I chose for this dissertation project is Neo4J. It is a main stream graph database which can incorporate a SPARQL interface and other important capabilities specifically useful in this project. Neo4J is utilized for the experimental approaches pursued and for visualization of the results provided here.

The main reason graph databases are especially useful for Recommender Systems has to do with how the data is stored. Graph databases give equal prominence to storing both the data and the relationships between them. In a graph database, we don't have to live with the semantically poor data model and expensive, unpredictable joins from the relational world. Instead, graph databases support many named, directed relationships between entities or nodes which gives a rich semantic context for the data. And queries are quite fast since there is no join penalty in a graph query as have with SQL.

Retrieving information from a graph database is called a traversal of the database. It involves walking across the edges of a graph to find objects which are connected. In a graph database, a traversal is a fundamental operation for data access. A major difference from the SQL query is that a traversal is localized and there is no global index as each node stores an index of nodes connected to it. The size of the graph does not have a performance impact on a traversal the way it does in a SQL join. There is a global index in a graph database like Neo4J. But, it only provides a starting point from which to start a traversal operation. To determine if a particular element has a given property, it does require a linear scan of all the elements which has a higher cost computationally than with an index which has a much lower computational cost. This trade-off does occur between graph and SQL which needs to be carefully considered depending on what one is attempting to do computationally. [66]

Between graph databases there is no standardization in the languages as in SQL which is quite standardized with some variations. The lack of standardization has led to many implementations and frameworks such as Neo4J. Cypher attempts to use a more keyword oriented system to be more SQL-like. So, this lack of standardization requires learning many approaches to graphs in order to select the best tool to solve a given graph problem. Neo4J has tended to become somewhat of an industry standard in recent years due to its clear structure and high adoption rate. Neo4J uses Cypher which to be more of a graph query language to avoid writing traversal code.

A particular strength of graph databases over the old RDBMS approach is for application in recommendations from association. Finding indirect associations from direct connections is particularly aided with the use of graph databases where associations are the focus of the query. Therefore, graph databases applications offer increased ability to find both the direct and indirect associations within a Document Network such as Semantic Medline.

Reliability of data storage in a graph database is also of great importance. Many of the SQL databases like Oracle and Microsoft have established and highly reliable database engine environments to protect the reliability of the data. For graph databases without this history, what guarantees the data integrity? In graph databases there is a developed standard now called ACID (atomicity, consistency, isolation, durability) which is a set of properties to guaranteed transaction processing reliability for graph databases like Neo4J. This reliability

aspect is extremely important if building mission critical systems on a graph database. Neo4J is ACID compliant and therefore, production ready like Oracle for the enterprise. So, Neo4J is now considered an enterprise ready application. [66]

High availability and security are also very much included in products like Neo4J as has long been available in the RDBMS counterparts. Neo4J follows master-slave design architecture for coordination and replication which can lead to inconsistencies for a short period of time which the data is not immediately synchronized. But, it utilizes Zookeeper to coordinate the nodes to keep them in sync which resolves the risk of consistency very well. Security features are also very well developed now in enterprise versions like Neo4J. And, for databases like Semantic Medline and other biological types of databases, it is a great fit because of its inherent structure. So, for all of these reasons, use for Semantic Medline is very well justified to see how Neo4J can improve on the capabilities now with the current implementation of Semantic Medline using MySQL. Performance can be much better if implemented properly.

Associations of Documents in a Document Network

Association rules mining is one of the dominate methods for data mining. These association rules reveal similarities between users for web pages which are utilized in Recommender Systems. They recommend web pages that appear to be useful for a given user. But, there are limitations of these association rules which can lead to loss of vital information potentially relevant for the users.

Typically, these association rules focus on co-occurrence within a given transaction set. These rules use hyperlinks to list “hard” connections which result from the hyperlinks of co-occurrence. But, this approach may avoid relationships between documents or pages. These “hard” connections expose direct relationships between documents or pages.

What about “indirect” relationships between the documents in a DN or pages from a web search? These are not exposed by traditional “direct” association rules. [65] Other researchers have called these “indirect” associations as “Semi-metric” connections; i.e. associations which do not conform to metric or Euclidean rules. [62]

There are a number of indirect or “transitive” association rules which can lead to important indirect associations not typically exposed by just using the “direct” rules. A “partial indirect” association rule is the indirect relationship between two objects with respect to one of those objects for which two direct association rules exists. The document or page in the partial indirect association rule is call the “transitive” document or page. Another indirect association rule is the set of all possible transitive document or pages for which indirect association rules between two objects exists. The complete indirect association rule aggregates all of the partial association rules between two objects with respect to all transitive documents or pages and is called the “complete indirect” association. A complete indirect association rule between two objects exists only if there is at least one partial indirect association. These complete indirect

associations are again NOT symmetric and are also known as “Semi-metric” associations.

In order to maximize the value of searches such as in Semantic Medline, these indirect associations need to be included as well in order to provide a complete response to important queries especially in the medical arena where discovery of new associations and relationships is extremely important. The use of complex association rules which make use of both direct and indirect association rules for the recommendation do greatly improve the results. These complex association rules have been explored to some extent in the literature. But, the use of them is somewhat limited in currently used Recommender Systems. I shed more light on them here. And, this does tie into the narrative having to do with Semantic searches and the use of context in these searches. The context is embedded in the association rules as it turns out and gets factored into the overall methodology based on the pre-processing of the data to compute the association distances within the semantic context of Semantic Medline.

For the Semantic Medline database which first searches by a concept to build a resulting set prior to inclusion of the predications, we need an indirect association rule and algorithm to include indirect documents in the first result set. Then, when the predications for this search result group are constructed, we need to bring in all the indirect predications for the original search group and for the predications themselves because there may also be indirect co-occurrence results of value. Then, the summarization and visualization can be done from

there. We need to experiment with optional approaches to conduct this two-step association pertinent for Semantic Medline and find the best method for this to occur. And, see what kinds of results get added to the results when approaching with indirect associations as well as direct to see what is happening by expanding the search approach.

Proximity and Distance Graphs Calculation Methods

One of the advantages of using the knowledge contexts in the recommendation architecture is that the same key terms can be associated independently between different information resources. Indeed, the distance functions of knowledge contexts allow us to regard these as connected concepts. This way, the same set of key terms describing the present interests (or search) of a user, is associated with different sets of other key terms in distinct knowledge contexts. Thus, the interests of the user are also context-dependent when several information resources are at stake.

Equation 1 - Proximity Calculation [62]

$$KSP\left(k_i, k_j\right) = \frac{\sum_{k=1}^{m}\left(a_{i,k} \wedge a_{j,k}\right)}{\sum_{k=1}^{m}\left(a_{i,k} \vee a_{j,k}\right)} = \frac{N_{\cap}\left(k_i, k_j\right)}{N\left(k_i\right) + N\left(k_j\right) - N_{\cap}\left(k_i, k_j\right)} \quad \textbf{(1)}$$

KSP is the semantic proximity between two keywords, ki and kj. It is the probability that the keywords co-index the same document in a DN whose semantics are defined by matrix A. The value depends on the document sets

indexed by the keywords and the intersection of these sets of documents. N(Ki) is the number of documents indexed by Ki, and N(ki,kj) is the number of documents both keywords index. So, from the formula for proximity, the keywords are close if they index many of the same documents. [62]

Computing Distance Functions: Associative Semantics

One very useful measure of association between documents in the DN is Proximity. It is actually a measure of co-occurrence between documents. A Proximity Graph is a graph obtained by computing the proximity for documents in the DN to use the proximity measure as the length of edges to connect the document nodes. Proximity graphs can be seen as associative knowledge networks that represent how often items co-occur in the large set of documents. The understanding is that the items which frequently co-occur are associated with a common concept which is understood by the users and authors of those documents. The graph of co-occurrence proximity captures the document network associations because we expect those concepts or themes to be organized in inter-connected subgraphs or clusters in the proximity network graph. [62]

These Proximity Graphs are reflexive and symmetric fuzzy graphs where we can perform transitive closure of these graphs. We can also transform the Proximity measure to distance where the edge weights denote dissimilarity represented by distance between the nodes. Short edges mean smaller distance where there is greater similarity. Long edges imply less similarity.

A high value for proximity means that the two items from one set of objects tend to co-occur frequently in another set of objects. But, what happens when items do NOT occur frequently with one another but do occur frequently with the same other objects. If they co-occur frequently with a third (or more) objects, should we infer that the two items have indirect associations as one might expect from the transitive property of associations, for example? We would expect objects with strong indirect relationships to be more associated than those which have weak indirect relationships. [62]

Transitivity can be more intuitively viewed by converting proximity to distance. The shortest distance between two objects may not be the direct edges between them but rather through an indirect path. These distance functions which violate the triangle inequality are referred to as Semi-Metrics. The transitivity may be violated which then defines Semi-Metric to be non-Euclidean. A metric graph would show the Euclidean distance. Semi-Metric includes those which are indirectly related and therefore, not metric or Euclidean. [62]

Semi-metric behavior is a matter of degree. For some pairs of objects in a distance graph, the indirect path may provide a shorter distance than other paths. To measure the degree of semi-metric behavior, another measure is used for semi-metric average ratios. [63] To compare semi-metric behavior between different DN and their respective objects, a relative semi-metric ration is used.

The semantic proximity between keywords is the probability that the keywords co-index the same document in a DN. The two keywords are close if

they tend to index many of the same documents. It uses the total number of documents indexed by the keywords on their own along with the total where both keyword pairs index the same document as an intersection set. [62]

Distance is calculated this way:

Equation 2 - Distance Calculation [62]

$$d\left(k_i, k_j\right) = \frac{1}{KSP\left(k_i, k_j\right)} - 1$$

d is the distance function which defines a weighted graph called the distance graph whose vertices are all the keywords extracted from a given DN whose edges are values of *d(k,kj).* A small distance between keywords implies a strong semantic relationship between the keywords.[62]

Knowledge Context

The collection of relevant associative distance graphs which can be extracted from a DN expresses the knowledge it conveys to the community of users as an information resource. In the case of Semantic Medline, this is a very large set of keywords, predications, and documents. But, each resource is organized differently and leads to different associative semantics tailored to a particular group of users with a varying history of use. The same set of keywords and predications can be related to different sets of documents resulting in different distances for the same keyword or predication pairs. We can refer to the relational information or associative semantics of each information resource as a Knowledge Context. This implies that the way documents are organized in

information resources expresses knowledge provided to their community of users. The knowledge context conveys some of the collective knowledge shared by the community of users for these resources. The distance graphs relating the DN define an associative semantics to convey the strength of associated pairs of elements in the specific network from which they are available. [62]

Specifically, we characterize an information resource *R* by a structure named Knowledge Context:

Equation 3 Knowledge Context [62]

$$KN_R = \{\boldsymbol{X}, \boldsymbol{\mathcal{R}}, \boldsymbol{d}\}$$

Where **X** is a set of available sets of elements *Xi*, e.g. **X** = {*K*,*M*, *U*}, where *K* is a set of keywords, *M* a set of documents, and *U* a set of users. **R** is a set of available relations amongst the sets in **X**, e.g. **R** = {**C**(*M*, *M*), **A**(*K*, *M*)}, where **C** denotes a citation relation between the elements of the set of documents, and **A** a semantic relation between documents and keywords such as a keyword-document matrix. Finally, **d** is a set of distance functions applicable to some subset of relations in **R**, e.g. **d** = {*dk*}, where *dk* is a distance between keywords such as the one defined by formula above. The application of these distance functions results on distance graphs *D* whose vertices are elements from the sets in **X**. Within the recommendation architecture from this dissertation, users are themselves characterized as information resources, where **X** may contain, among other application-specific elements, the sets of documents previously retrieved by the user and their associated keyword. The end result of what feeds

the recommendation algorithms in our architecture is the distance functions **d** of knowledge contexts.[62]

This gives us the context for the hypothesis testing with the Medline Database in which we compute the relationship distances from the Semantic Medline data and use it to create solution sets in order to explore the transitive property and how it can be used to find interesting relationships from the Medline database. Distance graphs provide the knowledge context of the Document Network. And, we attempt to visualize those graphically to compare to the direct connection graphs inherent from the base graph representation.

I can do the same thing with the Semantic Predications. Apply these rules there to the proximity between the Semantic predications and not just the keywords. I do both in this research where I get the keyword proximity and the Semantic Predication proximity. I use both to explore the distance between the documents and it represents an approach no one has implemented before. This approach is a new way to include context into the equation where a concept is slightly different between users who perform the search.

As an example of how Semantic Medline works now, please find in the following figure a case for Melatonin as the subject concept. And, the predicate is “TREATS” with all the objects found through direct connection in Semantic Medline using the direct connected objects as follows:

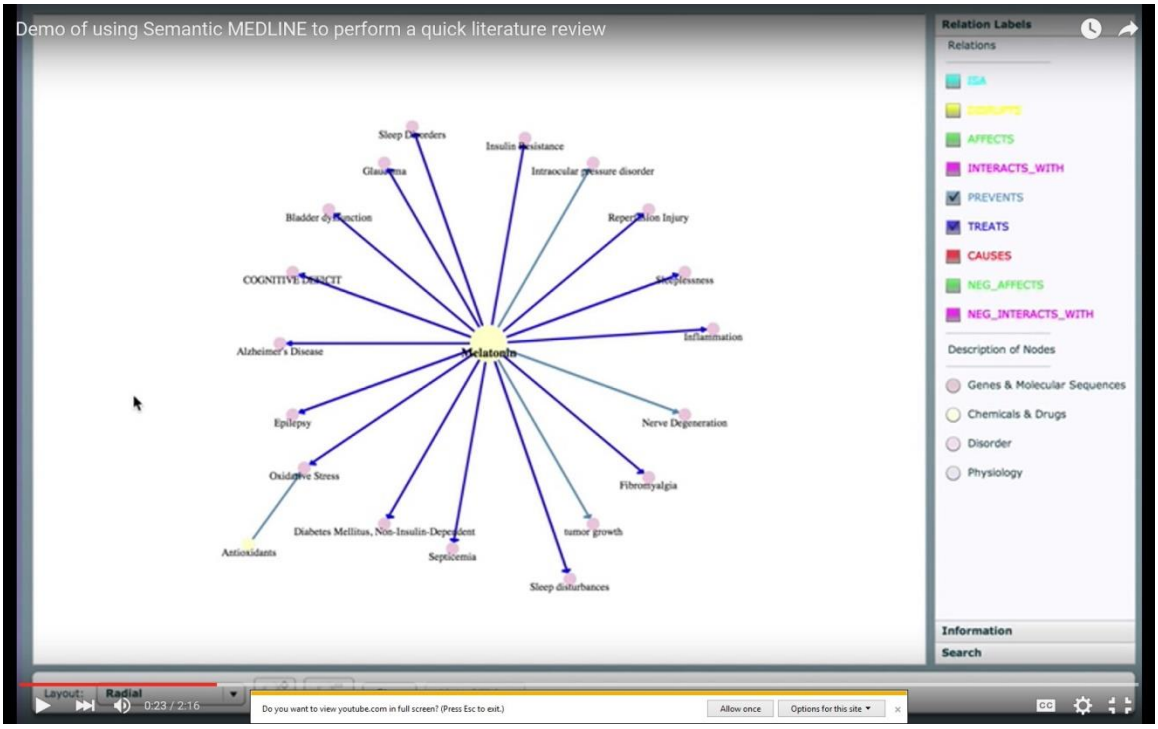


Figure 8 - Semantic Medline Output for Melatonin TREATS

Now to show what is done here in this dissertation for the same example, please find the next figure which shows the direct and indirectly connected objects for the same search terms “Melatonin TREATS” but this one is from the system created in this dissertation.

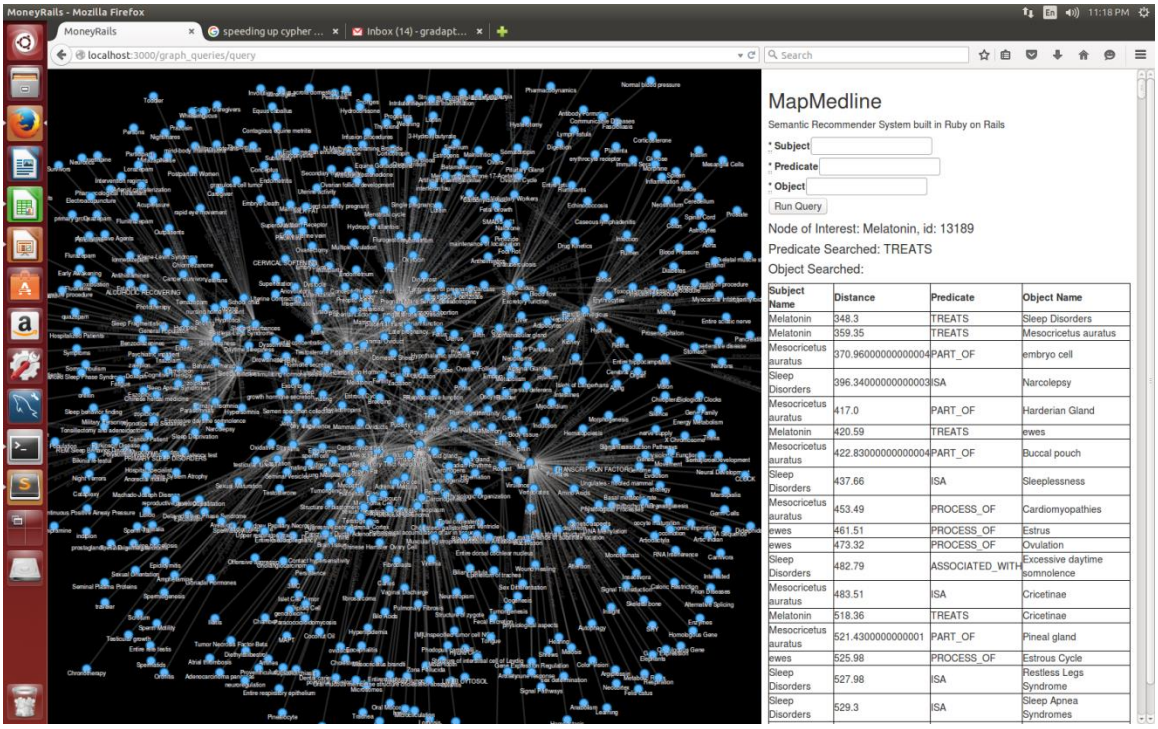


Figure 9 - New Semi-Metric Visualization of Melatonin TREATS

You can easily see where this dissertation is headed to get vastly improved results making use of distance calculations in hybrid, optimized Recommender System algorithms for Document Networks.

Recommender Systems for Document Networks

With this basis in the type of recommender system for consideration, now we need to find applicable approaches and understand how these may be implemented. These approaches are based in an algorithmic method. Several algorithmic approaches are recommended for document networks [63]. We look at these and recommend improvements using Semantic Medline and the

Semantic Predication as opposed to just keyword co-occurrence at the heart of the Rocha studies [62] and [63].These algorithms are based on 3 types of graphs:

1. Item-Based Proximity
2. Item-Based Semi-Metric Proximity
3. User-Based Proximity

I discover how these different approaches work for Semantic Medline and Semi-Metric using Semantic Predications as the objects. Also, I find new algorithmic approaches to work well as recommenders for documents in this DN.

With the Semi-Metric approach, I discover new relationships from the indirect associations not previously discovered from previous methods. These new relationships are implied by global associative semantics but not by previously retrieved documents that are known to be directly related. The newly discovered semi-metric pairs can fill a gap for novel documents implied by the DN. With just a metric distance function, I would not discover these implied associations. The Semi-metric approach offers the potential for new recommendations not known previously. An example would be useful to discuss to illustrate the concept of Semi-Metric. Let us assume we have the following measures for similarity between 3 objects (A, B, and C) as follows:

- Jaccard Similarity(A,B) = 0.5 (50% of features in common)
- Jaccard Similarity(B,C) = 0.5 (50% of features in common)

- Jaccard Similarity(A,C) = 0 (no common features, no overlap at all)
- Distance(A,B) = (1/0.5) - 1 = 1
- Distance(B,C) = 1
- Distance(A,C) = INFINITE

This case violates idea of the triangle inequality. In other words, just because A and B are related and B and C are related, A and C may have absolutely no relationship and we really can't infer the distance from A to C as being less than or equal to the sum of A to B plus B to C. But, there is a greater probability that A and C do have some relationship had this indirect relationship not existed. These kinds of indirect connections need to be explored in case a direct relationship may exist. This probability of indirect relationships is included in the recommender algorithms optimized in this dissertation.

Architectural approaches for a ranking algorithm approach to DN Recommender Systems will be needed for a solution. One such approach is called IDARM (Indirect Association Rules Miner). This is the second stage of the recommendation process after the direct associations have been mined. The general concept of IDARM is as follows: [64]

Input

1. L1 - Draw a set of direct associations where the confidence is greater than a given value
2. L IR- List the indirect associations with their confidences

3. L T - List the numbers of transitive associations for each indirect association

Output

1. Full list of indirect associations – L IR
2. Full list of numbers of transitive associations – L T
3. Full list of confidence numbers associated with each association to rank associations

Using this ranking approach, we can provide new knowledge in some cases while in others it may be more about confirming existing connections or knowledge. The more transitive triads which exist in the set of indirect relationships, the more likely there is a positive contribution for the indirect associations. [64]

The graphic below shows the architecture for an IDARM system which could be implemented for a Recommender for Document Networks. The important aspect is the component parts required to provide such an approach. The direct and indirect miners are treated separately. Then, there is a merge capability to bring the two lists together and rank them prior to presentation. This is a fundamental architecture used for Document Network Recommenders. [64]

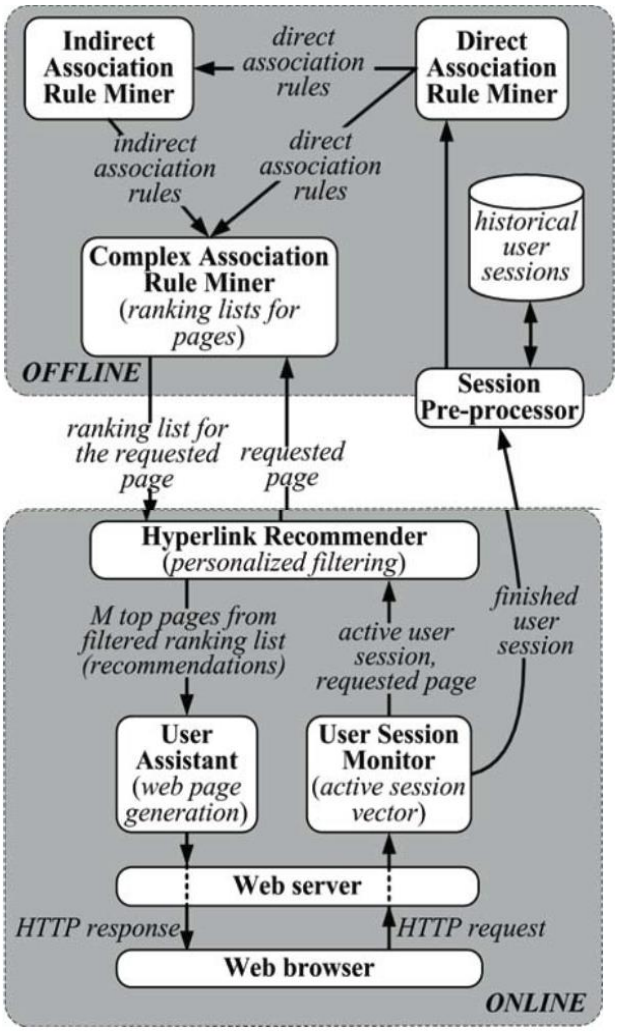


Figure 10 - IDARM Technical Architecture [64]

IDARM System Architecture

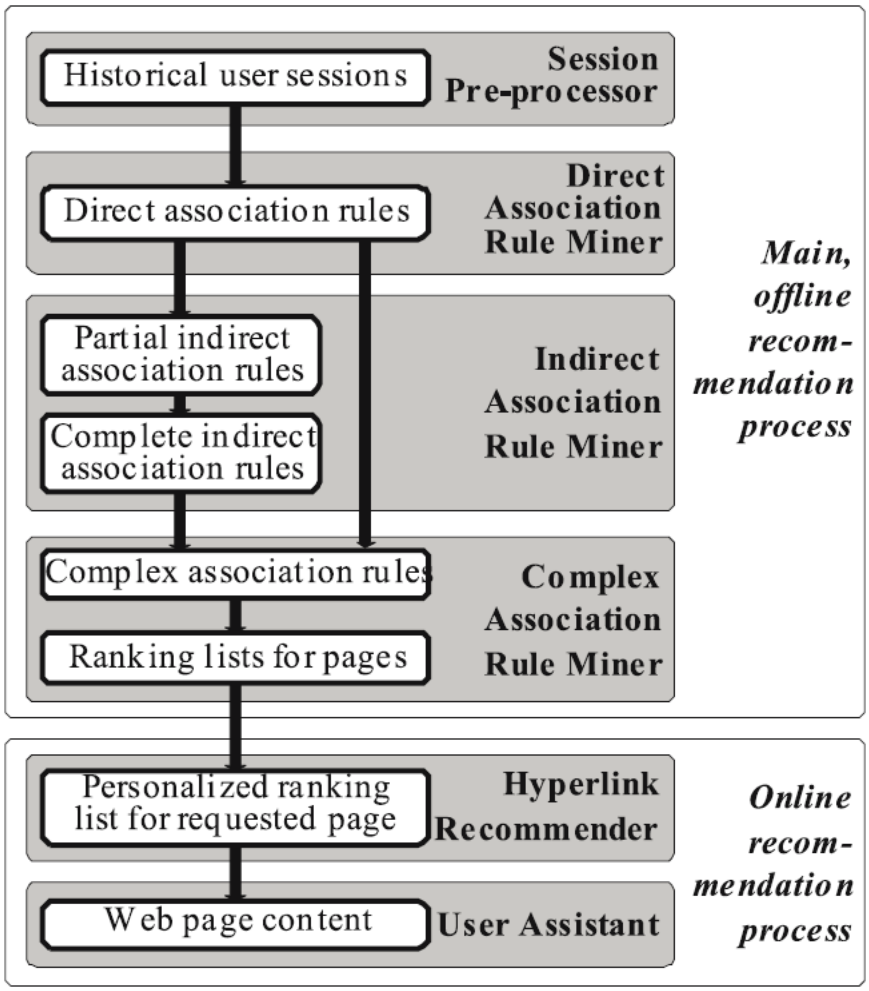


Figure 11 - IDARM Functional Architecture [64]

These two architectural diagrams actually offer a comprehensive and reasonable way to implement a recommender system for Document Networks in general. There is nothing here which is not easily recognizable from a development perspective. They also include the user specific aspects which are very important for the incorporation of context and user shared interests. The

algorithmic approaches employed in this dissertation will make use this architectural approach for implementation. I adapt it here for our experimental framework to test optional algorithmic approaches using the some of the blocks in the flow chart shown in the overall architectural diagrams above. [64]

Proximity Algorithms

Considerable research has been performed in the area of Proximity functions to measure associations. Proximity of nodes in a graph is defined in terms of the percentage of closeness between two nodes where related nodes have a value of 1 and unrelated nodes have a value of 0. An RDF graph as in the case of Semantic Medline is actually a typed multigraph which means that any pair of nodes can be connected by several edges known as properties. Proximity is a function of the edges between the nodes and not the nodes themselves. And, proximity is a function of the paths between nodes which may consist of multiple edges. With this in mind, proximity functions take on a form based upon the path within the graph for how the proximity is measured. Proximity must take into account all the paths between nodes and therefore relies on an infinite series mathematical approach. Shorter paths do have a higher weight than longer paths.

These proximity algorithms expand each path in the given set of paths using the set of edges using the end of a path to expand to the next new node. And, they expand the size of the paths and not the cardinality of the sets of nodes by an amount equal to n + 1 where n is the size of the original path set.

This is path expansion approach is important to have in mind for algorithms which measure the association of nodes in a graph RDF database representing documents in a Document Network. I provide a discussion of path expansion as a very important approach for graphs to represent interests. There are a variety of ways to do this and those are embedded in the experimental options selected for analysis. [68]

Distance for Indirect Transitive Closure Edges

I have reviewed indirect associations and found these could be very interesting for the recommendation and ranking of candidate citations in response to a given request. I have discussed methods for computing the proximity and then the distance between nodes for these connections. And, I have discussed ways to find the indirect edges to provide candidates to consider for the recommendation based on indirect techniques either from semi-metric or indirect associations.

But, how do we compute a distance for the indirect paths so they could be ranked in the recommendation list? So far, we have not considered this and this becomes an important consideration once indirect associations are exposed. Now I discuss this aspect and try to find a reasonable way to make this kind of calculation based on what we know at this point in terms of the direct association distances and the indirect edges.

Most of the research for complex graph networks will treat these interactions as binary edges in graphs even though in a real situation the

interactions have a wide degree of intensity and strength. [69] So, more recently, the research has begun to focus on real complex networks as weighted graphs. This is a relatively new shift in focus and much is left to do in this area. Past research has shown that pairs of items for which there is no direct co-occurrence information but which are strongly related through indirect paths will possess higher probability of co-occurrence in future networks. In such weighted graphs, there is an infinite number of ways to compute transitive closure to compute indirect associations in the data. This means there will be different forms of transitivity of indirect association in complex networks modeled as weighted graphs.

Recommender approach using Proximity Graphs

Based on the previous discussion regarding proximity and distance, how can we utilize this information in a Recommender algorithm to select and rank items which result from this approach? So, we can consider 4 kinds of Recommender algorithms to use these graphs as follows:

1. **Item-Based proximity**
    a. Retrieve the user vector which contains the set of associated items from the training set R.
    b. From the item-based proximity graph, remove columns associated with items that do not appear in the user profile.

c. Calculate the mean value of row weights for each row in the reduced item-based proximity graph matrix from step 2. This provides a scaler score for all the items in the matrix.
d. Then, present the items recommended for the top n scored items.

2. Item-Based Semi-Metric

a. Same as Item-Based Proximity except the item-based proximity graph is enhanced with additional edges.
b. Calculate the metric closure from the proximity isomorphism equation.
c. From the resulting distance graph, identify the semi-metric pairs (edges) with below average ratio above the threshold, and insert the corresponding edges from the transitive closure of the item-based Proximity graph into the original Proximity graph.
d. Then, we use this Item-Based proximity graph and use the steps in algorithm 1 above for the Item-Based Proximity.

3. User-based proximity

a. Determine the k nearest users to the current user from the User-based Proximity graph which are the k highest values.
b. Recommend the top n most frequent items among the neighborhood of users

4. User-Based Semi-Metric

a. Here we enhance the proximity of the User-based proximity graph with the semi-metric edges just like in algorithm 2 above. Then, we use algorithm 3 above.
b. For both semi-metric algorithms (2 and 4), the thresholds for the below average ratio were set based on the distribution of the ratios around the cut-off point of the power law.

These options give us some reasonable ways to make use of the semi-metric information and graphs in order to actually provide recommendations. Then, we would evaluate which of the above approaches provides the best recommendations for the given user. [63]

Algorithmic approach for Medline Case Study

What is the best way to evaluate the success of the project for improvement of the recommendation approach for documents in Semantic Medline? The Appendix also has a great deal of information on factors for consideration for evaluation of recommender systems. This material clearly states that the evaluation approach does depend on the type of recommender you are working with. Clearly, a document network is an item type recommender system. Supervised methods are not appropriate because of the requirement to have experts to supervise the classification process. I need unsupervised approaches which are accurate and easy to run on their own as you might find with a web browser with page ranking. Clustering, of course, is an unsupervised approach. But, how would you cluster a document network to support a

recommender application? Actually, this is a difficult question to answer definitively because it would depend on the goals of the recommender system. The Appendix provides a great deal of detail on many ways to implement Recommender Systems and their methods for Evaluation. So, how to cluster would depend very much on the goals sought.

As was discussed in the introduction of this dissertation, I am interested in finding the best approach to provide document citations to a user based on their interest profile established through previous interactions with the recommender system and based on declared interests. With filtering systems, the function of a filter is to establish the relevance value of each document according to the interests of the user profile and then present citations for documents to the user based on the filtering. Filtering can involve grouping, sorting, pruning, and rank ordering document citations based on relevance values. So, from a machine learning perspective, this becomes a classification issue to group documents according to classes if a supervised method is available. If unsupervised, then clusters are more appropriate. There are many classification and clustering schemes. As presented previously, Semantic Medline database is constructed with the RDF triple predication which combines concepts with predicates and objects using the MESH medical language structure which begins the classification process by using medical language standards for grouping. Then, the predications need to be classified as well according to their associations with other documents in the document system. This additional level of classification at

the predication level helps to establish a better fit to the user profile than with just concepts alone. Semantic Medline by virtue of the supervised learning approach with MESH provides an abundant and reliable source of pre-classified training data for classification of the predications.

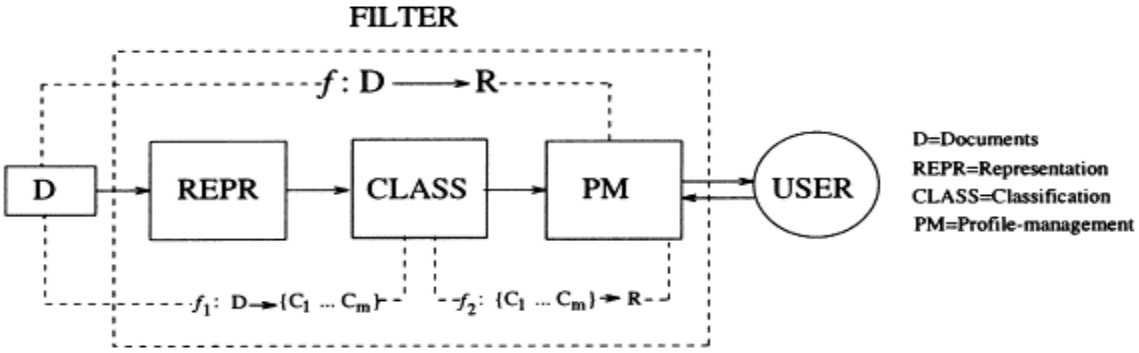


Figure 12 - Filter Method for Recommender Systems [70]

The above figure provides a scheme for document representation, classification, and interest profile management which is highly modular where virtually any technique applicable for each block in the architecture can be integrated into a single system. Our goal here in this dissertation is to look at a number of techniques to determine the best ways to accomplish for our selected case study using Semantic Medline.

Historically, this kind of filtering process for document network recommendations has been associated with a vector approach. For the representation step, the document vector is where each concept represented in

the document receives a weight. Each document vector length is based on the total number of tokens in the thesaurus, and each element in the document vector receives the weight corresponding to the tokens and augmented with the frequency for the concept element appearing in the document. Then, for classification, the classes are also identified by documents with comparable vectors. Since it is similar to a neural network classification approach with weighted elements, then neural network algorithms are useful to support this vector approach to filtering documents in a document network like Semantic Medline. But, of course, as was stated earlier, there are many other ways to do this especially with graph databases which should have greater potential accuracy and efficiency in a Recommender System. [70]

The Neural Network approach requires the availability of a training set with known classifications for each case. In some document recommender systems, the document relevance is only available after the search and can serve to reinforce the classification. For those situations, a clustering approach can group data into similar groups. This kind of unsupervised approach requires little human intervention as pre-classified documents are not needed. Semantic Medline has incorporated classification schemes both supervised and unsupervised and the results are contained in the predications available for further clustering or classifications. Already embedded in Semantic Medline, we have a very high level of classification included in the data we are using.

To evaluate classification quality, classical information retrieval performance measures may be used with the following parameters:

a. Total number of documents classified into a class that agree with an expert's judgment
b. Total number of documents classified into a class that do not agree with expert judgement
c. Total number of documents not classified into a class that according to an expert should belong to a class.
d. Total number of documents not classified into a class that according to an expert do not belong to a class.
e. All of these parameters do depend on an expert's judgement.

This leads to measures as follows:

a. Recall = a/(a+c) – the proportion of class members as determined by the expert that were accurately placed in the class
b. Precision = a/(a+b) – the proportion of documents placed in the system's class that are accurately placed.
c. Error rate = (b+c)/(a+b+c+d) – it includes both the errors of commission and of omission.

These parameters give us a way to judge the quality of classification from a numerical perspective if I apply a classification approach such as neural networks and vector classification. To do this though, I do need an expert to know the quality of the resulting classification process. [70]

One previous study which used Medline as the document network and used a neural network approach to classification had some interesting results to keep in mind for our case study in this dissertation. The classification performance demonstrated high level of variance in terms of recall, precision, and error rate. A single measurement alone did not capture the true classification capacity of a given class. The tree measure approach was better equipped to provide the classification measures needed. In some cases, recall could be high but the precision could be relatively low and vice versa. The ability of a class to attract correct documents while avoid incorrect ones will improve both recall and precision. The class capacity was well captured with the error rate measure. Classes with low error rate demonstrated higher and more similar recall and precision. And, classes with high error rate had lower and more dissimilar recall and precision. These measures proved to be very useful against Medline documents. [70]

Also, from the same study, classification performance influenced the resulting filtering as one would predict. Higher classification accuracy did provide improved filtering performance and conversely, poor classification accuracy did lead to degraded performance in filtering. Also, supervised classification approaches did lead to advantages in filtering as well when compared to unsupervised approaches. The unsupervised approach was more open ended in terms of the types of classes generated. Clustering algorithms used for the unsupervised approaches lead to classes which varied greatly in terms of scope

from narrow to very broad. The supervised approach had more tightly managed class scope. Supervised approaches took more time and resources but lead to improved filtering results overall. [70]

I want to find better methods now and take this to the next level by also using our experimental framework with Python and Neo4J to see where I can improve the performance using the predication as well which come from Semantic Medline. The filtering studies did not use Semantic Medline with the predications. I want to use other algorithms than neural networks which have better proven results for graph databases like Semantic Medline applying it to the predications in addition to the concepts which were the subject of these previous filtering studies. But, these previous filtering studies do give us a method for evaluation and specific parameters to use for comparison purposes which is utilized in the experimental section of the research. It is useful to utilize this information for looking at new approaches for the recommendation process to take it to another more accurate level. Now I discuss the analysis of other algorithms and approaches for Recommender Systems by looking at a number of other options building on these filtering studies.

Ranking Algorithms for Document Recommendations

Web page ranking is similar capabilities to those needed for our DN database approach. We are basically in need of the best way to rank document citations by their similarity to a concept defined by the user. This is much the same type of method used in web page ranking methods.

For example, an algorithm for page ranking is given as follows: [78]

PR(A) = E(A) (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))

Where

PR(A) is the PageRank of page A,

PR(Ti) is the PageRank of pages Ti which link to page A,

C(Ti) is the number of outbound links on page Ti and

d is a damping factor which can be set between 0 and 1.

First of all, we see that PageRank does not rank web sites as a whole, but is determined for each page individually. Further, the PageRank of page A is recursively defined by the Page-Ranks of those pages which link to page A. Ok! This seems like a decent way to actually pre-rank documents individually and then build a rank for a given concept query from there based on the documents linked to key documents that support a given recommendation query.

Also, inherent to this approach, PageRank of pages Ti which link to page A does not influence the PageRank of page A uniformly. Within the PageRank algorithm, the PageRank of a page T is always weighted by the number of outbound links C(T) on page T. This means that the more outbound links a page T has, the less page A benefits from a link to it on page T. Again, this is something like we need for document recommendations. The weighted PageRank of pages Ti is then added up. The outcome of this is that an additional inbound link for page A always increase page A's PageRank. Finally, the sum of the weighted PageRanks of all pages Ti is multiplied with a damping factor d

which can be set between 0 and 1. Thereby, the extent of PageRank benefit for a page by another page linking to it is reduced. [78]

How does this idea of page ranking apply to documents in a document network? The rank is influenced by the number of links to a given page for sure. This would be analogous to the number of edges to a specific citation when representing the document network in a graph database with edges or links. There are strong analogies here with our Semantic Medline case in which concepts and predications tie to objects through edges. In our case, I look at it more as the proximity or distance between the documents and I find ways to measure those distances or similarities. Other references discuss these linkages as Similarity Measures. It is easy to see these linkage concepts are very much one in the same and ranking is the objective for recommender systems dealing with document networks just as with web page ranking.

Page ranking is an iterative process and it builds to a more accurate number as pages are ranked. The iterative process is a simple linear stationary process. At each iteration, a Page Rank vector (single 1xn vector) holding all the page rank values is computed. These vectors are a hyperlink n x n matrix H and a 1xn row vector represent the probabilities for the connected pages or documents in our case where the probability is a normalized probability representation using the distance calculation. H is a row normalized hyperlink matrix with a value when there is a link between nodes and 0 otherwise. It is the classical power method applied to matrix H. H looks like a stochastic transition

probability matrix for a Markov chain. The dangling nodes in the graph create the zero rows in the matrix. [78]

In a Markov chain, we know the rows of the matrix are the inputs and the columns are the outputs given the row heading as the input. The values in the matrix are the probability the input of the row heading produces the output column heading. And, the values in a given row all sum to 1 as the probability of all possible outcomes are 1. This is very similar to taking our distances or other measure of similarity and normalizing them to a probability which sums to 1. Then, the Markov chain matrix can be built and proceed with the algorithm through transitions by multiplying by the transition matrix possibly a number of times, one for each iteration. The actual probability is found in the matrix after transition by iteration. These values can then be inserted in the page-ranking formula to rank the documents. This is basically how I need to create the process in our experimental approach discussed later. It help us to yield a document ranking which is quick to run and which improves with transactions as the values continue to improve through iterations. And, the current value needs to be stored with the predication in our case for the next time when it is iterated again similar to how I stored distances now with the predication for future use to cut down on execution time. I experiment with Page Rank for optimizing the search algorithms for this dissertation and adapt them significantly to the Semantic Medline case. [78]

Another ranking algorithm which has received considerable attention especially for document networks is called WICER for Weighted Inter-Cluster Edge Ranking for Clustered Graphs. This approach would imply the existence of available clusters. I have presented methods for building clusters in this research. And, I can now explore methods for tying WICER into the page-ranking algorithm to combine the strengths of both for a Document Network. So, let's now explore more about WICER and how it works for possible inclusion. [76]

First, it is useful to identify another algorithm which is used broadly for page-ranking on the web called HITS – Hypertext Induced Topic Selection algorithm. In general, HITS and Page-Ranking ranks nodes in a graph with directed edges. But, when natural clusters exist within the graph, because of the added importance of nodes belonging to specific clusters, these algorithms do not capture semantic information of the clusters to produce an efficient ranking of the nodes. WICER is an algorithm to rank nodes in a clustered graph which could very well be a Document Network. It includes a parameter used to value the inter-graph edge weight which weighs edges between different clusters more than edges within the same cluster. There is another parameter used to weigh the nodes based on the number of different kinds of edges that connect to it. These parameters are used to rank the nodes in the graph. [76]

HITS and Page-Rank give uniform importance to all the nodes and edges in the graph. However, in something like a Document Network like Semantic Medline, we have semantic information which describes the type of node or type

of edge. In this type of graph, the nodes can be clustered and the edges categorized based on this semantic information. And, this new information can be used to provide a new ranking scheme that utilizes the types of nodes and edges. [76]

It is interesting to consider an example. A medical researcher searching for a medical reference related to a specific medical case would be very interested in a document that is referenced by multiple types of cases rather than a document being referenced by similar cases. Therefore, it would be logical to assign such a document a higher rank in the search results. Also, it is more likely the medical researcher who may be an expert would already know about similar cases possibly. The WICER algorithm basic ideas are:

a. A node that has incoming inter-cluster edges should be ranked higher than a node that has incoming intra-cluster edges
b. The rank of a particular node is weighted by the number of different clusters from which there exists an incoming edge to this node, and
c. Each cluster is weighted based on its importance.

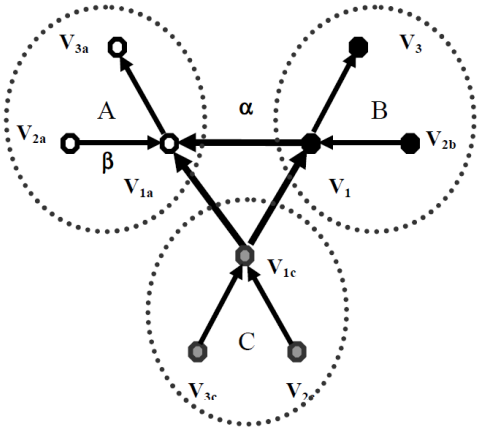


Figure 13 - Edge Rank Clusters [76]

Equation 4 - Edge Rank Formula [76]

The Weighted Inter-Cluster Edge Rank is given by,

$$R(V_{ic}) = \frac{(1-d)}{N} + d * \left(1 + \frac{C}{N_c}\right) * \left(\sum_{j=1..C} W(j) * \sum \frac{R(V_{kj})}{OD(V_{kj})} * w_{jc}\right)$$

Where,

$R(V_{ic})$ is the rank of vertex V_i of cluster c.

N is the number of nodes in the graph.

N_c is the number of clusters in the graph.

d is the damping factor to handle rank sinks.

$W(j)$ is the weight of cluster j.

C is the set of clusters that have an edge to cluster c.

S_j is the set of vertices in cluster j having inlinks to vertex V_i of cluster c.

w_{jc} is the weight of the edge from cluster j to c.

$w_{jc} = \alpha$, if j ≠ c

$w_{jc} = \beta$, if j = c

α is the inter-cluster edge weight

β is the intra-cluster edge weight

The parameters Alpha and Beta are the inter-cluster and intra-cluster edge weights presented earlier which can be valued based on the document cluster and the semantic significance within the Semantic Medline document network. The pseudo-code for this method is as follows:

```
Function WICER( in G : Directed Graph with N nodes
                        Nc : Number of clusters, α : Inter-Cluster Edge Weight,  β : Intra-Cluster Edge Weight
                        W : Weights of Nc clusters)
return R[1..N] : Rank values of the nodes
prevR[1..N] : Temporary storage of rank values
for ( i = 1 to N)
   R[i] = 1/N;
RSink : Set of nodes those have zero OutDegree
while (diff > ε)
  for ( i = 1 to N)
        c : cluster to which Node i belongs
        Compute the rank of Node i
        for (j = 1 to C)
              R(Vic)= R(Vic) + W(j)*∑k∈SjprevR(Vkj)*wjc/OutDegree(Vkj)
    R(Vic)=(1-d)/N + d *R(Vic);
        // Handle Dangling nodes with OutDegree = 0
        HandleRankSinks (RSink);
  end for
  diff = | R - prevR |
end while
```

Figure 14 - Edge Rank Pseudo Code [76]

In experiments with different types of graphs, the WICER approach once significantly adapted to Semantic Medline does perform very well against plain page-ranking approaches because it combines in the additional elements associated with the semantics of the network being ranked. It is clearly a very important additional algorithmic approach to consider here for Semantic Medline which offers the benefits of the clustered approach with the page-ranking approach all in one algorithm. We consider it in our options for experimentation

and see how the use of the principals provided with WICER can help in our case. [76]

Link Prediction Algorithms

Understanding the association between two specific nodes raises interesting concerns such as:

What are the factors that drive these associations?

How is the association between two nodes affected by other nodes?

An interesting problem is how to predict the association between when there may be no associations between these nodes in the current state of the graph such as is the case with semi-metric, indirect nodes we have discussed previously. This is the link prediction problem. It is related to inferring missing links from the observed graph network where based on observable data, we try to infer additional links which may not be directly visible but which are likely to exist. This is a problem which comes out of social networks and edge ranking already discussed. A good example of it is Facebook “Friend Finder” which attempts to connect people who are not currently connected but whom may be pre-disposed to similar interests and therefore be ripe for connection with such algorithms. Of course in our case study for medical references with Semantic Medline, it is more a matter of link prediction to find interactions between nodes such as proteins through the medical literature. Link prediction could be used to accelerate research connections and collaborations that would take longer to form on their own. **[77]**

In terms of the graph model, the problem becomes given a graph structure as I have in our case study for Semantic Medline in which an edge represents a connection between citations that are available currently in the database, then the link prediction task is given this network to then output a list of edges not present in the current graph that are predicted to appear in the network. The current graph would be used for training purposes for the predictive model.

To generate such a list of edges, a heuristic algorithm is used to assign a similarity matrix whose entries represent scores between the nodes from the training data. This score between nodes is viewed as a measure of similarity between the nodes. All of the non-existent nodes are sorted in decreasing order according to their scores. The links at the top are the ones most likely to exist.

To test the accuracy of the link prediction algorithm a large fraction of the observed links (90%) from the known graph are used as the training set. The remaining 10% links are used as the probe set to be used to test the prediction capability of the algorithm which is interpreted as the probability that a randomly chosen missing link from the probe set is given a higher score than a randomly chosen nonexistent link. The degree to which the probability exceeds 50% indicates the link has a better chance than just pure chance. The more the score exceeds pure chance then, of course, the higher is the chance of existing and becomes a predicted link. **[77]**

We saw from the discussion of semi-metric and indirect connections that the sum of the links to an indirect connection can be finite. However, the inferred

connection from a node to the indirect node may be non-Euclidean and not be less than the sum of the component parts. There could be an infinite distance to the indirect node or it could be a connection which is less than the sum of the parts and be Euclidean. We cannot actually derive a distance from this semi-metric analysis. But, with link prediction algorithms, we can predict the probability of the connection score using machine learning techniques of training the model and then using it to predict the probability. **[77]**

This would be extremely valuable in the case of Semantic Medline if a good model can be trained and used quickly when a new research query is presented to the search algorithms. And, it would be able to help us rank the indirect nodes much better than we can now as well. It would be a big improvement over edge ranking as presented previously and provide more accurate predictions potentially. However, it does require training and pre-processing to accomplish clearly. Facebook does the “Friends Finder” feature with background threads to be presented to the user at some point after having been compiled for the user. This too could be accomplished for Semantic Medline assuming users are created with profiles and background threads are created to be run and presented next time the users logs in to the system.

We have already discussed measures for similarity which could be employed in link prediction algorithms. There are a number of other measures as well which have promise. These are as follows:

Common Neighbors: computed in the context of collaborative networks to verify a positive correlation between the number of common neighbors of two nodes and the probability that the nodes will collaborate in the future.

Jaccard's Coefficient: measures the probability that two nodes have a common feature which is a good measure for the similarity of the nodes.

Frequency-Weighted Common Neighbors: measure based on number of common features between the nodes where rare features have higher weighting.

Preferential Attachment: measure based on concept that nodes with many connections more likely to have new connections in the future.

Exponentially Damped Path Counts: measure that sums over a collection of edges which is exponentially damped by the distance to count shorter edges more heavily. The more edges between two nodes the stronger the connection.

Hitting Time: a random walk starts at one node and iteratively moves to a neighboring node chosen at random. The Hitting time is the number of steps to move from one node to another.

Rooted PageRank: based on page rank described earlier here but adapted to link prediction. It is a ranking of overall importance which biases toward topically relevant and marked pages.

Some of these approaches are more node based such as Common Neighbors, Jaccard's Coefficient, Frequency-Weighted Common Neighbors, and

Preferential Attachment. The others listed above are more edge based. The node based algorithms have restricted scalability and are not as viable for user generated content networks. Computing features on a subgraph only of nodes is computationally intensive. However, edge based approaches are more scalable and are less computationally intensive. These approaches lend themselves better to document networks like Semantic Medline. **[77]**

Building on these concepts, it would be important to find technical models for our platform with Semantic Medline to implement some of these kinds of approaches particularly as with the edge based approaches. Since we have already worked with PageRank and Edge Rank, these can be utilized to help predict links as well as help to build a ranking score I use in the actual ranking processes. And, it turns out there are a number of articles available for these algorithms being developed in Python which could be implemented in our technical environment for evaluation as threads running in the background to make predictions when a user is searching for specific relationships. But, the threads once launched may not actually finish during the session in process and may present results afterwards when the user logs in later since they can be time consuming to run completely.

Concerns with Algorithmic Query Approach Options

I have considered many approaches to this point for use in the Semantic Medline query for providing an improved Recommender System. There are more discussed in the Appendix. I have listed some ideas below which may have value

to select the best approach. A combination of these approaches is the best way to take the advantages of some and eliminate the disadvantages.

1. Train a model of Medline searches by the choices made once inside of the search tool. Then, use that for next user to show the areas of greatest interest when combining in the indirect links. Store the sequence of entries as the user performs the searches. Then, when a new user starts with the same search, recommend based on the stored searches what other users have found most interesting. **Problem** – With the size of the Semantic Medline database and number of users, this could lead to a huge storage problem not to mention the execution time which would be required for this kind of approach.
2. Experiment with different ways to branch out to more indirect cases with 4 entries for example – subject – predicate- object – Subject – predicate – object. Go 2 predications deep with the search program. Store the entries even if stop before enter 6 entries – Use those to build an interest model. Provide a selection to show the previous choices or default the next entry for the user. Show visualizations of the past searches from model. **Problem** – Again, execution time and storage when have such a large database with exponential connections.

3. Clustering algorithms use distance to build clusters with shortest distances such as K-means. Once build these clusters, then use them for the Recommender if a selection falls within a cluster. For this, can just build an algorithm to run through all the nodes to build the clusters. **Problem –** Clustering many relationships and training a model for new entries is a massive undertaking with a large and complex system like Semantic Medline.
4. Select nodes to traverse down based on thresholds for the next node. The thresholds are determined by SVM techniques. Use a formula for the threshold as arbitrary and affected by the user model of the searches performed. It is based on the distance away from the target node. **Problem** – Vectorization of subgraphs can take on a difficult life of its own and again lead to very long execution times.

The bottom line is that many of these approaches are nice in theory but for a Recommender System to be responsive to users, it has to act something like a search engine page ranking approach to be effective. Complex learning schemes are unwieldy for a very large graph database like Semantic Medline.

CHAPTER THREE

Experimental Platform and Approach

One of the main best recommendations from the research is computing the distances based on the frequency of predication occurrence in the Medline database. In order to do this, I set up an environment in which I could test the computational methods for best results. First, I have to add a distance or weight to every relationship in our network at least when the query is executed. Each distance is computed between two nodes and depends on the connectivity of each.

Once the weight field is added to each edge's properties I can query the resulting Neo4J database with a search query that finds a concept node based on the search term. The graph is traversed from this node with a breadth first search to find the sub-graph with a max-level. This max-level should be at least 2, more results and perhaps finer resolution can be achieved with a larger max level. This max level is the number of edges from the source node that is returned. If A -> B and B->C a max level of 2 returns all of the B and C nodes but not D if C->D.

Here I am interested in the outgoing relationships i.e. I return B if A->B but not C if A<-C. This limits the results and produces more meaningful behavior.

Once I create a subgraph that has the results from the BFS, I can find the shortest path lengths based on all of the edges. I also get the paths on which these lengths occur for example: BRCA1 -> INHIBITS -> LAN61 -> AFFECTS -> Lung Cancer

Distance Calculation and Algorithms

The distance is computed using a sophisticated frequency calculation based on the connectivity of independent nodes. [62]

Here I find the equation:

KSP = N_i_j/(N_i + N_j - N_i_j)

where

N_i : Sum of "count" of all relationships of node i

N_j : Sum of "count" of all relationships of node j

N_i_j : "Count" of relationship of interest

The KSP is a quantity that represents the strength of the edge of interest relative to the other connections of each node. For example a gene "BRCA1" is relatively well known but not as broad of a term as "Breast Cancer" which it is related to. Here the N_i_j quantity is large but N_j : the quantity of total predications including "Breast Cancer" is much larger just due to the amount of publications including the concepts. Thus, the distance is represented by: [62]

d = 1/KSP - 1

This distance is much larger if separating these two terms and then running the distance calculation. This technique presents novel connections when compared with techniques that do not include relative frequency calculations. [62]

However, this technique requires that these distances be calculated a priori so that searches can be conducted at web speed. The starting database is created based off of counts of each predication occurrence. This count field is stored in the edge properties. Thus, to transform the count into a distance we need the above pieces of information. The resulting distance is then added to the relationship as a property.

In order to compute the distance of each edge, we must traverse the full graph network. This is accomplished by first querying all of the relationship ids in the database (neo4j). Then we loop over each relationship, querying the id for the full structure, and calculating the distance. To decrease traversal time I use a cache to store the N_i or "Sum of all counts" for each node in a dictionary. The compute distance function only executes a database query to determine this quantity if this node and quantity is not found in the hash table. Further speed-ups could possibly be achieved here. This is important to further develop since new publications are added to MEDLINE frequently and the latest semantic information is important to analyze. Thus, I must improve the speed of our distance calculation to support some monthly update.

The algorithms to query the Neo4j graph once the distances have been pre-processed and stored inside of the database are then managed directly by the database itself. The experimental code supplies the cypher query structure to conduct the query and then handle the visualization.

In order to make queries run quicker, I run a pre-processing step to calculate the N_i (counts of nodes) and N_i_j (count of predications) in advance so these are available to perform the distance calculations already without searching for all the locations where these occur. Those pre-processing steps to take a while to execute and could easily be done when new citations are provided to the Semantic Medline environment. This pre-processing aspect is critical to getting recommendations quickly for real time processing requirements.

Technical Platform

The experimental platform was created using Python and Neo4J on a Windows platform and Ruby on a Linux platform also using Neo4J. PyCharm contains a group of Python programs which can query the Medline Neo4J graph database. PyCharm is the Python IDE environment for code development and trial execution. The results are sent to Chrome for display after a query.

Since there are so many records in the Medline database, it was important to narrow the scope of the database load to Neo to the predication or the triples where the relationship is not to the node itself. The predication must be to another node. In this case, the nodes are concepts. I loaded all the node concepts where there are relationships to other nodes. And, I dropped all the

nodes which do not have predications to other nodes in this narrowed database. I felt the predications contain the information of most interest and I needed to have fewer records to work with because of the response times with millions of records otherwise loaded in Neo4J.

Several modules were developed to configure the environment, display the results in Chrome, query the NEO4J, build the distances, and to sort and rank the output once the query is executed. Github is used for versioning. Google Drive is used to share files between computers. Have several systems being used. One is with Quadcore I7, 16GB of memory, 256G SSD, 1TB hard disk, and NVIDIA GPU for high performance processing using Windows Server 2012. So, since have a few systems, to keep everything in sync using various open source software to keep everything in sync. The Python and Github screens look like this:

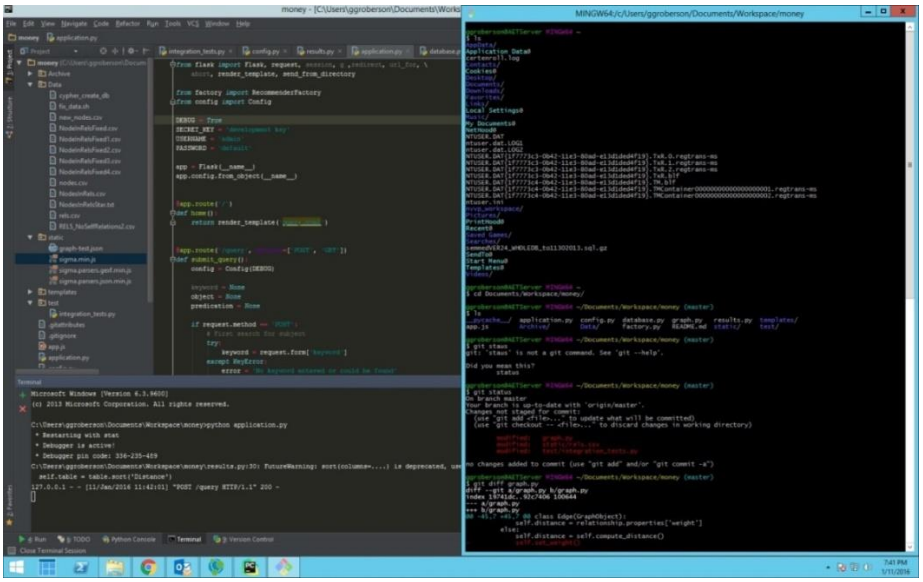


Figure 15 - Python Experimental Platform IDE

Python with Chrome output screen appears as follows:

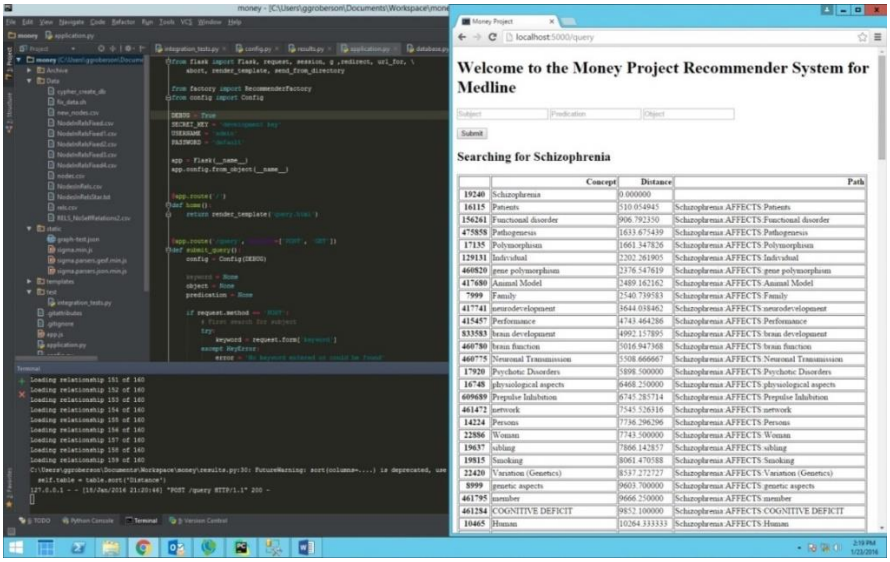


Figure 16 - Python Search Screen Example

Shown in the Chrome output is a sample search with Schizophrenia as the concept and all of the associated predicates and objects are shown from Neo4J query, filter, and sorting by distance (smallest to largest ordering on distance).

To show that I am running Neo4J here, the screen showing the database running is here with the Python and Chrome minimized:

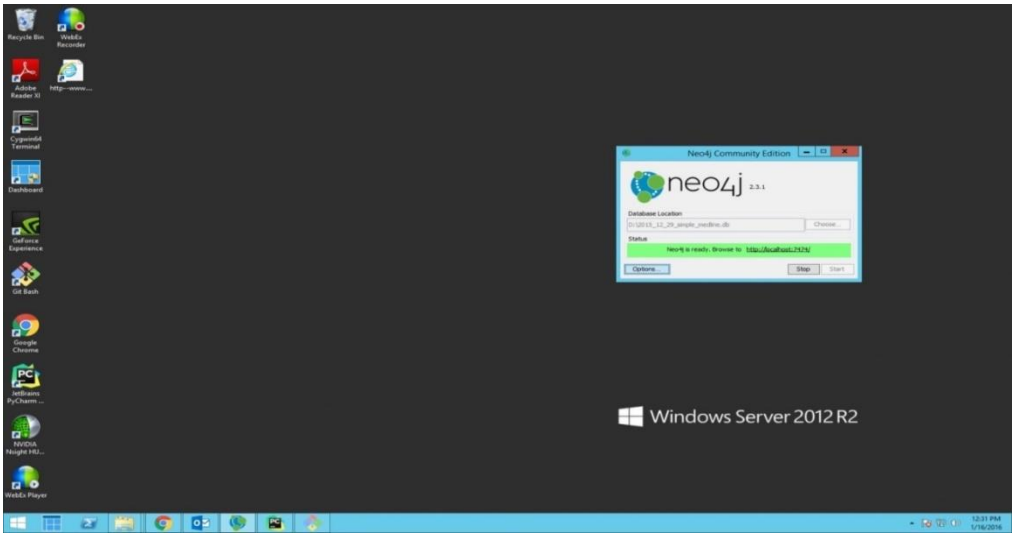


Figure 17 - Neo4J Running

An example of the visualizations I got from NEO4J on Medline with the gene BRAC1 is here:

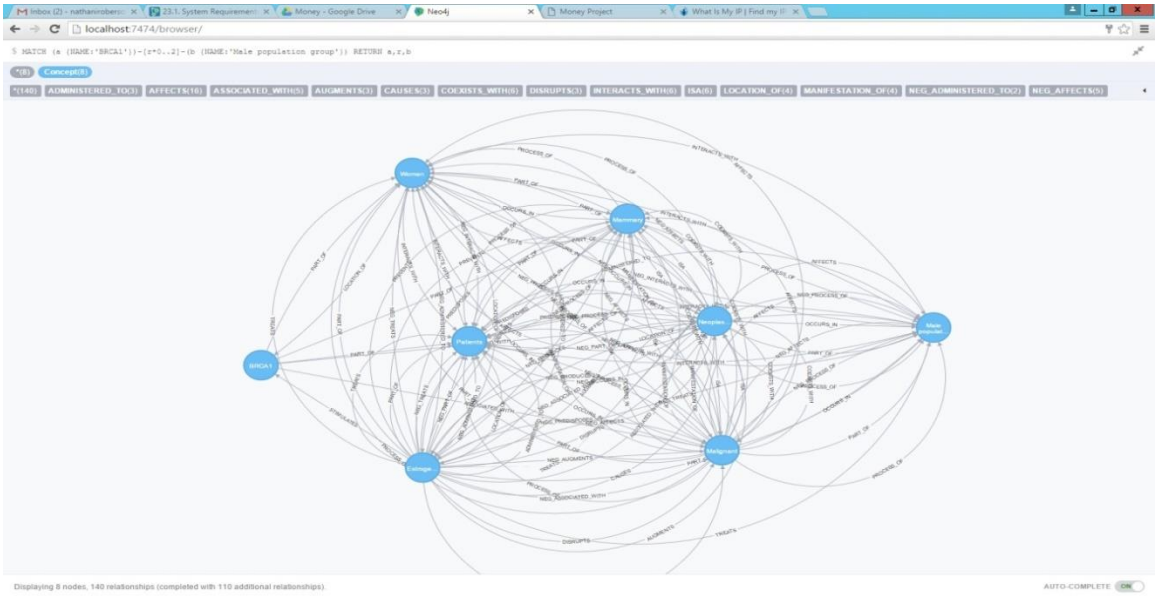


Figure 18 - BRCA1 Example Visualization

This graph visualization is the associations of the concepts with self-associations eliminated. Gives a quick look at the complexity of the concept relationships found in Medline.

DN Recommender Solution Algorithm

The architecture for this solution is as shown here:

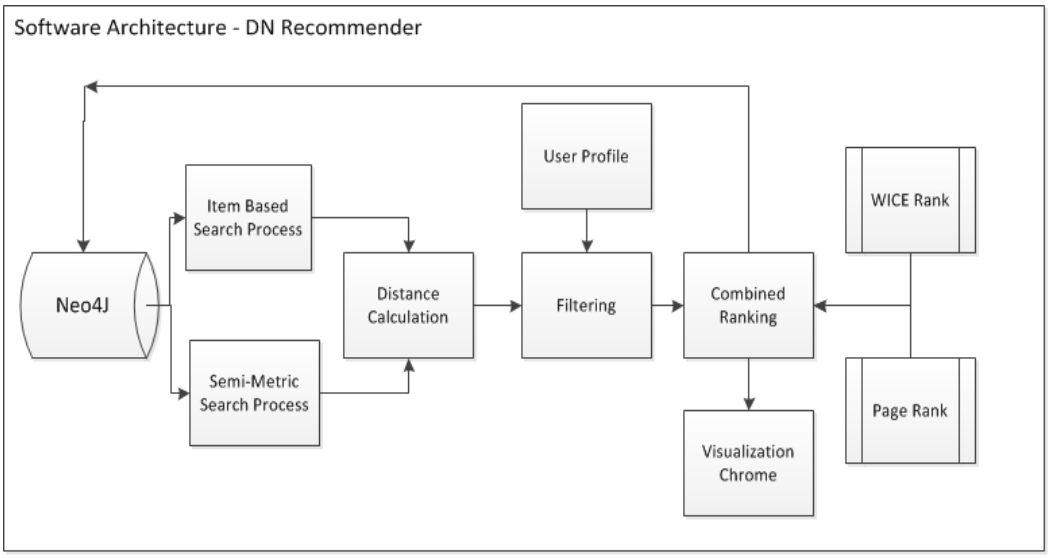


Figure 19 - DN Recommender Solution Architecture

The following steps are followed within the DN Algorithm to recommend citations to the user based on their entered search criteria:

- From the search terms, cast a wide net using direct and indirect associations to build an initial selection set of links.
- Read the persisted counts for use in distance calculations.
- Calculate the association distance to the search terms as the principal measure to use for final ranking.
- Read the Page Rank and Edge rank stored values to adjust the ranking of the selection set of links when the values have significance based on past activity.
- Filter the selection set with the user profile of preferences to cull items from the final listing when the items are not of interest.

- Build a final ranked selection set for display using the previous calculations and combine them in such a way to optimize their performance based on test results.
- Display the final ranked selection set in both graph visualized and table format available for further analysis and filtering.
- Update the Page and Edge ranking persisted values using the appropriate formulas based on final selection set in order to capture the new activity.

This is how the solution algorithm works to find citations after being optimized.

I also built another infra-structure which was more Ruby on Rails running in Ubuntu Linux oriented to make use of existing Gem algorithms which are available for Neo4J to see how that would respond as well. This architecture is more in the direction of the previously referenced IDARM (Indirect Association Rules Miner) approach which enables the entire system to be easily made available through internet access to a web server. The algorithms between the 2 environments are very similar but implemented in two different infra-structures to evaluate the differences for scaled implementation. This environment enables the use of pre-established and optimized Neo4J query tools which actually made the construction much quicker and had the mining runs execute much quicker than with the python approach. But, python is more flexible and provides an ability to build your own algorithms a bit easier to test custom enhancements. Ruby on the other hand allows us to use Neo4J optimized queries which make this whole

thing run a great deal quicker however. And, with this huge database, Ruby is really the preferred way to go for that reason mostly. Everything takes so long to run, that it makes Ruby the preferred environment. So, both environments have strengths and weaknesses. I wanted to see how they responded for this research as part of the experimental process. A screen shot from the Ruby environment on Linux is shown here:

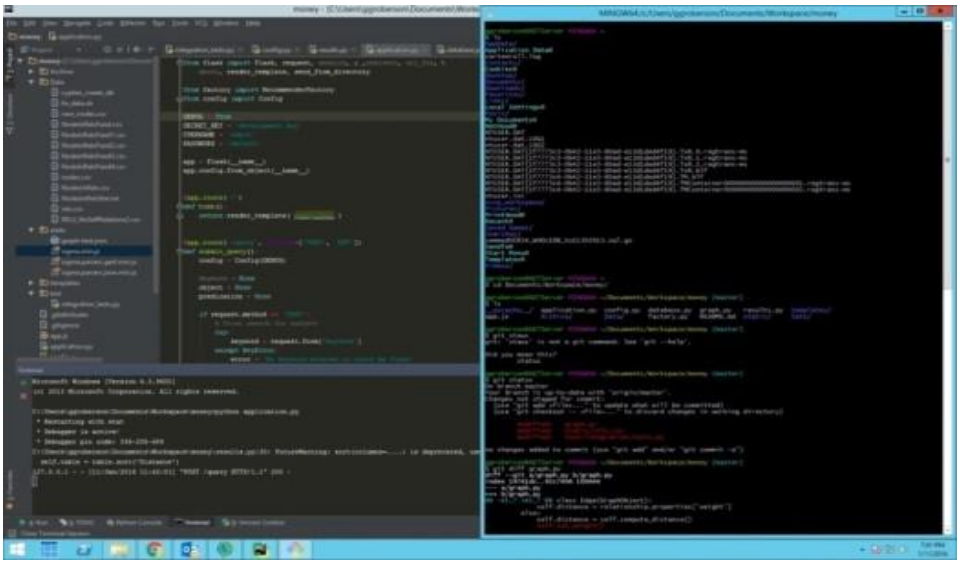


Figure 20 - Ruby Search Execution Environment

I run through an example user scenario to present the idea of the query approach with the Semantic Medline database in Neo4J using our Ruby query infra-structure. Once started, the system prompts the user as follows:

Figure 21 - Ruby Search Inputs

When I enter BRCA1 for Subject only, I get this result:

Subject Concept ID	Subject Concept Name	Distance	Predicate	Object Concept ID	Object Concept Name
1334573	BRCA1	157.0	ASSOCIATED_WITH	999811	Sporadic Breast Carcinoma
1334573	BRCA1	461.0	INTERACTS_WITH	55020	Deoxycholate
1334573	BRCA1	1667.0000000000002	LOCATION_OF	795985	Neural Stem Cell
1334573	BRCA1	2036.0	LOCATION_OF	415182	MCF7
1334573	BRCA1	3645.5	LOCATION_OF	1169347	Breast Cancer Cell
1334573	BRCA1	7249.500000000001	PART_OF	1115659	Mammary Neoplasms
1334573	BRCA1	20504.333333333332	STIMULATES	7647	Estrogens
1334573	BRCA1	25015.0	PART_OF	13699	Mitochondria
1334573	BRCA1	35137.0	ASSOCIATED_WITH	3019	Malignant neoplasm of breast
1334573	BRCA1	37985.0	LOCATION_OF	3782	Cell Line
1334573	BRCA1	83224.09999999999	PART_OF	22886	Woman
1334573	BRCA1	262509.0	ASSOCIATED_WITH	14388	Neoplasms
1334573	BRCA1	416128.5	LOCATION_OF	22886	Woman
1334573	BRCA1	1959318.2000000002	PART_OF	16115	Patients

Figure 22 - Ruby Sample Listing

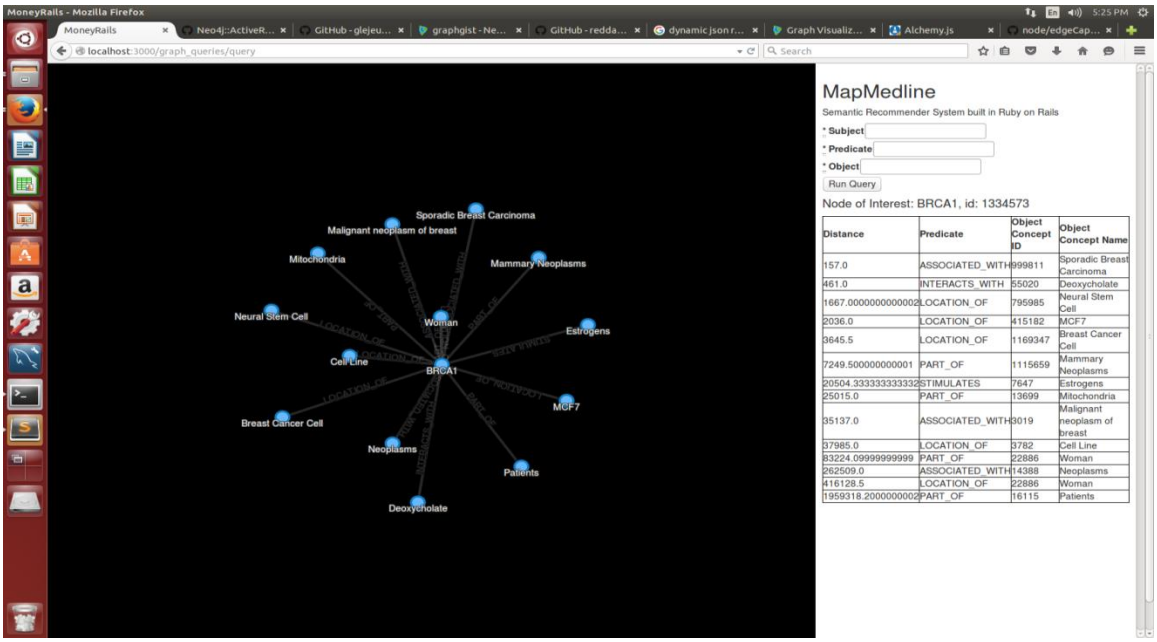


Figure 23 - Ruby Sample Subgraph Visualization

With the first entry being the opening predication to see what the gene BRACA1 may have predications with in the Semantic Medline database. And, I get these several relationships with the predications and objects. The semi-metric distance is calculated and the results are rank ordered by the distance as shown above in the response table.

Next, I may want to find out more about the “Associated_With” -> Sporadic Breast Carcinoma. I enter this full predication into the search screen to get the following result.

Subject Concept ID	Subject Concept Name	Distance	Predicate	Object Concept ID	Object Concept Name
1334573	BRCA1	22.0	ASSOCIATED_WITH	1334573	BRCA1
1334573	BRCA1	28.0	ASSOCIATED_WITH	1347685	C11orf30
1334573	BRCA1	91.0	ASSOCIATED_WITH	992261	chromosome 17q
1334573	BRCA1	111.0	PREDISPOSES	1335795	FANCD2
1334573	BRCA1	124.0	TREATS	98080	Bilateral mastectomy
1334573	BRCA1	131.33333333333334	COEXISTS_WITH	622996	Allelic Imbalance
1334573	BRCA1	157.0	ASSOCIATED_WITH	999811	Sporadic Breast Carcinoma
1334573	BRCA1	177.0	ASSOCIATED_WITH	1177182	11p15
1334573	BRCA1	181.5	AFFECTS	414933	Gene-Environment Interaction
1334573	BRCA1	186.0	ASSOCIATED_WITH	1177508	16q
1334573	BRCA1	386.5	PROCESS_OF	831854	Double Strand Break Repair
1334573	BRCA1	393.00000000000006	COEXISTS_WITH	637539	Genomic Instability
1334573	BRCA1	431.66666666666663	COEXISTS_WITH	366901	Somatic mutation
1334573	BRCA1	472.14285714285717	PROCESS_OF	13443	Methylation
1334573	BRCA1	487.99999999999994	PROCESS_OF	371013	Turkish population
1334573	BRCA1	523.0	COEXISTS_WITH	1167870	BRCA1 Mutation
1334573	BRCA1	574.6666666666666	COEXISTS_WITH	357173	Loss of Heterozygosity
1334573	BRCA1	724.8	COEXISTS_WITH	638192	Microsatellite Instability
1334573	BRCA1	882.9999999999999	ASSOCIATED_WITH	1337928	NFKB1
1334573	BRCA1	1010.125	AFFECTS	460820	gene polymorphism
1334573	BRCA1	1020.2941176470589	ASSOCIATED_WITH	51757	Tumor Suppressor Genes
1334573	BRCA1	1273.5	ASSOCIATED_WITH	5581	Cytosine
1334573	BRCA1	1572.0	AFFECTS	417675	Homologous Recombination
1334573	BRCA1	3606.0	TREATS	12984	Mastectomy
1334573	BRCA1	4987.5	AFFECTS	454296	Estrogen Receptor alpha
1334573	BRCA1	4987.5	ASSOCIATED_WITH	454296	Estrogen Receptor alpha
1334573	BRCA1	5276.833333333334	PART_OF	109967	Ovaro-
1334573	BRCA1	5474.5	LOCATION_OF	4369	Chromosomes
1334573	BRCA1	7324.5	COEXISTS_WITH	460123	Breast Carcinoma
1334573	BRCA1	7517.0	COEXISTS_WITH	4362	Chromosome abnormality

Figure 24 - Ruby Sample List Result

With this listing, we see in close proximity with smaller distance the relationship with C11orf30 gene. Next I look more closely at these associations since they are relatively close in distance. Searching now for just gene C11orf30, I see the following result:

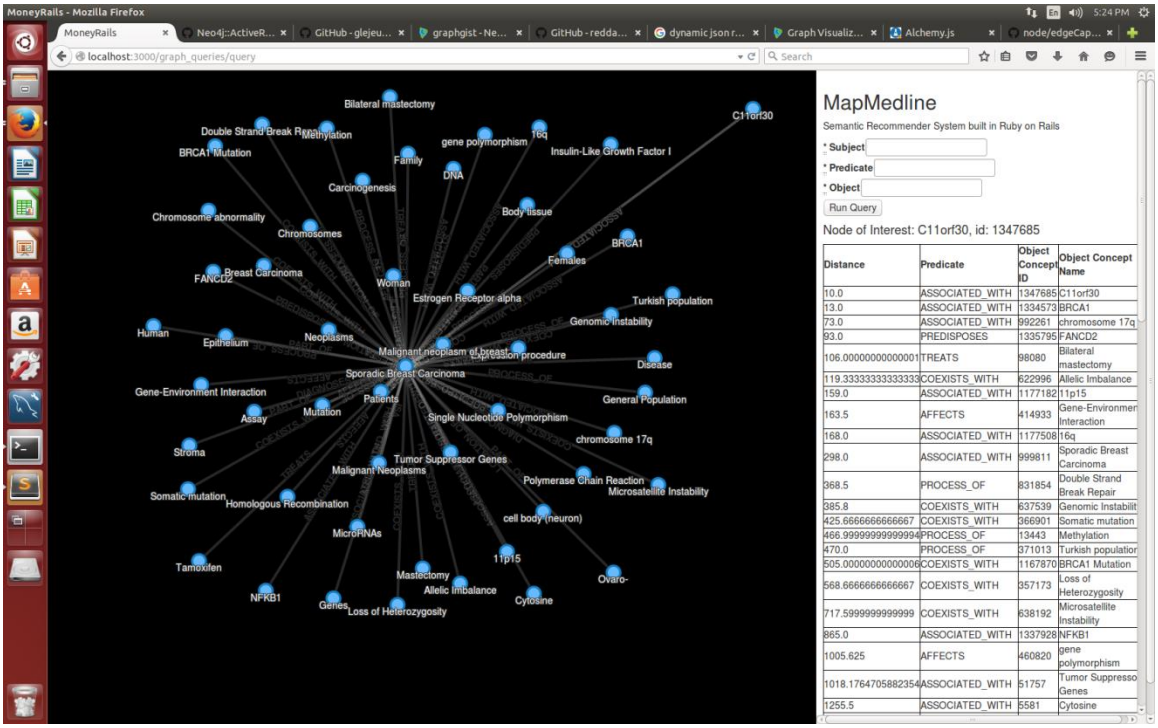


Figure 25 - c11orf30 Sample Ruby Visualization

We do not see here BRAC1 which is interesting. But, we do see close association through Sporadic Breast Carcinoma however. This is an excellent example of an indirect association which has become apparent with this type of mining approach. It seems like a reasonable conclusion from this search scenario presented by the screens shown above. Looking them up together in Google, we do find that C11orf30 is an amplifier for BRAC1. There is a reasonable validation then that they are strongly associated which would offer more research citations to recommend here.

Take a more complex concept like diabetes and we have to limit the number of predications to show in the graph to 1000 for this resulting subgraph:

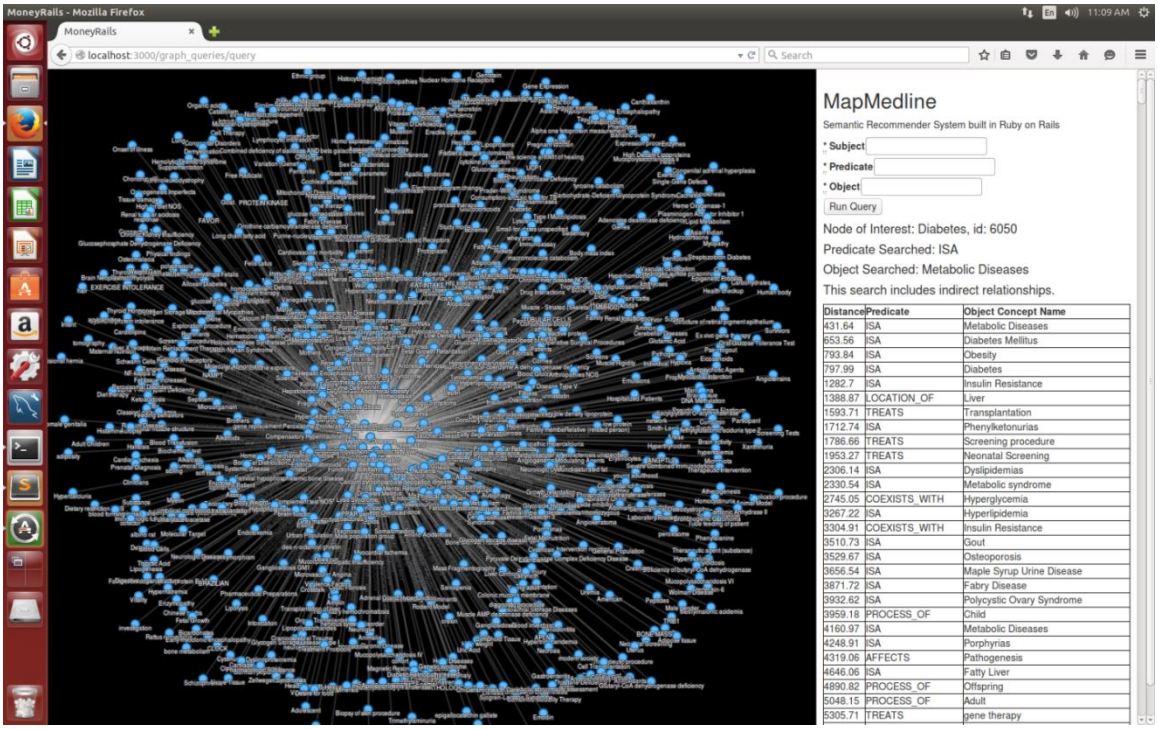


Figure 26 - Metabolic Diseases Sample Visualization

This shows the vast number of connections for something like diabetes and the huge base of documents which pertain. Again, the distance calculation is very useful to show which ones may be more closely associated here.

Now, let's take it to next level with a more refined approach to searching and displaying results for BRCA1. Instead of showing all the connections at just

the first level of connection, what would happen if we base the connections on the total distance from the core predication and build the resulting graph and table by the distance where additional connections when summed together could still be the next ranked recommendation to present. When I do this with any of our queries, I get many more lines in the resulting data set to sort and I do need to limit the number of lines in order to be able to present them in a reasonable length of time and within the memory available to the system. Now I do this for BRCA1 and get this resulting graph:

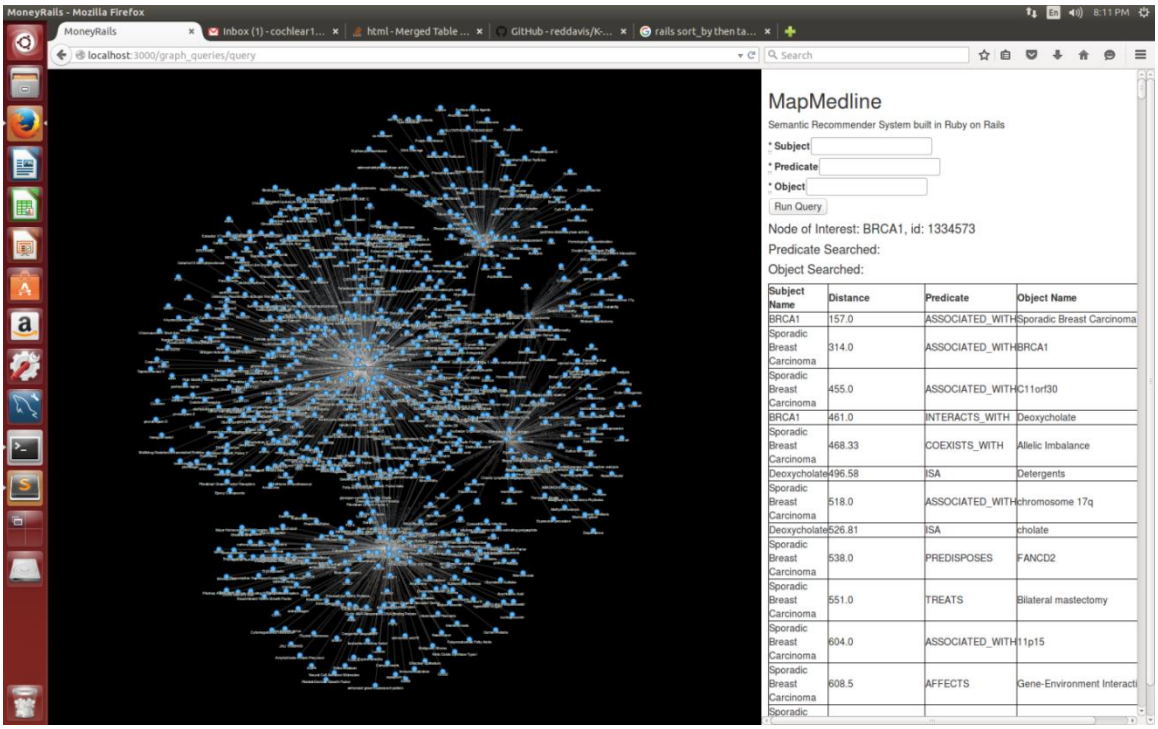


Figure 27 - BRCA1 Sample Ruby Visualization

I see now new lines in the resulting output which build off a connection to a 2nd degree connection which has a shorter distance than some in the 1st degree. One of those is BRCA1 “Interacts_With” Deoxycholate and Deoxycholate ISA Detergents and cholate further down in the list ranked by distance. This now is putting 2nd degree connections ahead of 1st degree if the distance is shorter which is exactly what I want to show for an accurate ranking. And, I now get a different level of indirect connections which are very novel for the BRCA1 gene which can now be explored by the user. These would not have been shown by existing Semantic Medline search approaches. I have a novel approach which brings to the fore potential new connections for analysis. However, I do run into severe memory issues with this approach and I do have to limit the list considerably as we get to these higher degree connections from a technical perspective.

It would be excellent now to produce this graph as a graph distance visualization to see the distances involved and understand the potential associations better with the predications included in the graph. And, it would be interesting to build the clusters for these concepts and see how the members of the clusters are related as well. This is just a glimpse as to what we can achieve with this kind of search capability. And, I am doing some of these suggested additional aspects here.

Validation Testing Methodology and Results

The testing is divided into validation testing and performance testing. Validation testing is performed to make sure the graph database was constructed properly and all the correct results are being realized prior to iterating the solution to get the best results with different optional solutions. Validation took the test cases prepared for the evaluation and first compares the direct connected items to the Semantic Medline results to make sure all these directly connected connections are appearing in the graph list results. Since the graph lists are easily exported to Excel as are the MySQL query lists prior to move to Neo4J, I was able to easily export the two lists for each test case and compare the results to make sure the new graph search results had all the same connections as did the MySQL searches. It took a few days to do all of them but I was able to make sure they were all there in the graph search without exception. With this information, we know the graph database search is doing at least as well as the current MySQL approach with Semantic Medline. Then, it becomes a matter of trying different algorithmic approaches to find a better approach beyond the search without enhancement to include 2nd degree connected indirect associations with their ranking. A baseline is established and the research knows concretely that the graph database is getting good results. Next, we prepare Performance tests to improve the process and iterate to a better solution.

Performance Evaluation Methodology

How do we evaluate our results using the Performance Evaluation analysis provided in the Appendix? We have a variety of choices and we need to pick the one approach which fits best for the graph type we are using and the optional approaches we are attempting with this research. Let's go back to our evaluation measures presented previously for this kind of graph database. Again, the measures are represented with the following parameters all based on being members of the query filter list resulting from the document query:

a. Total number of documents classified into a list that agree with an expert's judgment
b. Total number of documents classified into a list that do not agree with expert judgement
c. Total number of documents not classified into a list that according to an expert should belong to a list.
d. Total number of documents not classified into a list that according to an expert do not belong to a list.

All of these parameters do depend on an expert's judgement. This leads to measures as follows:

1. **Recall** = a/(a+c) – the proportion of list members as determined by the expert that were accurately placed in the list
2. **Precision** = a/(a+b) – the proportion of documents placed in the system's list that are accurately placed.

3. **Error rate** = (b+c)/(a+b+c+d) – it includes both the errors of commission and of omission.

An assessment of the experimental error was conducted to make sure the final results will be meaningful. The individual components or error were estimated and an overall calculation of system error was estimated. After the results were obtained, this inherent systematic error was compared with the error rate of the transactions to make sure results had substantial validity.

Next, I found the best ranking of subgraphs for example searches which represent the expert's best guess of the proper ranking and then measure how our optional approaches do with these measures. Then, I can compare the results from quantitative measures and find a best approach based on this evaluation method.

For this, I found several key medical concepts which were well known to some medical experts for whom I had access as family members. I used some key concepts they knew extremely well from experience and past research to prepare the best version of recommended document citation rankings using Semantic Medline with our technical platform since I have the entire database loaded and available for analysis. Then, I try various approaches covered in the preceding research to measure outcomes and compare results. From that experimental approach, I can pick the best approaches to recommend new methods for extending Semantic Medline search capabilities.

I also want to tie in the previous discussion for various algorithmic approaches now referring back to these 4 methods:

1. **Item-Based proximity**
2. **Item-Based Semi-Metric**
3. **User-based proximity**
4. **User-Based Semi-Metric**

The difference is that these approaches need to be modified to support a Recommender System environment supported mostly by ranking algorithms and less by trained machine learning trained sets of subgraphs stored for use with the recommender. The options I use are drawn from these 4 Recommender algorithmic approaches and I utilize optional methods for the creation of document ranking using our technical platform in which the code itself implements the optional approach. The combination of these optional approaches is measured with respect to their recall, precision and error rate. Then, put into a chart to yield a quantitative measure for overall comparison of results between the various options. This is the experimental part supported by months of research and construction of a technical platform for analysis of optional approaches using the full Semantic Medline database scaled down to just the predication aggregates but without the specific sentences which reference the document citations. There is a one-to-one relationship between sentences and the document citations. I am focused on the RDF which models the sentence knowledge. And, then, I infer the connection to the citation itself is

one-to-one and is a simple lookup. But, I am not going to actually provide the lookup in our experimental platform. It isn't necessary and it doesn't change the results at all.

The optional algorithmic approaches I use for testing are as follows:

- **Option 1** – **Modified Item-based Proximity** – Since we are focused more on a ranking algorithm approach for a Recommender System environment, the use of training sets is not appropriate. But, this option is based on item direct connections by Proximity or Distance as presented in previous research discussion.
- **Option 2 – Modified Item-based Semi-Metric –** This is very similar to Option 1 but with the inclusion of the Semi-Metric indirect relationships as well.
- **Option 3 – Modified User-based Proximity –** Again, since we are doing a Recommender System, it is more appropriate to have a user profile to support the user oriented filtering needed. And this combined with Option 1 should provide a better fit to the interests of the user.
- **Option 4 – Modified User-based Semi-Metric –** Same as Option 3 but include the Semi-Metric indirect connections like in Option 2.
- **Option 5 – 8 - Page Rank and WICER Edge Rank** features to create 4 more Optional approaches for each type of rank method. The essence of these ranking approaches were utilized but adapted to use for Semantic Medline with the graph database we had here. They remain self-adjusting

to continue to improve as the system is used to provide rankings which reflect user interests.

And, since the results could depend on the medical query chosen, I obtain results for a number of these queries and compare their results altogether for an analysis of how results may change with the query chosen. Hopefully, I find some commonality between the medical queries in terms of results.

User Profile for User-Based Options

In order to include user interests which extend beyond just the specific query being evaluated, I store a user based profile to include more details as to the specific interests of the medical research user. As I have found in our Appendix information for Recommender Systems, having such profiles is quite often needed in order to tie in additional user information. And, with this profile, the user can further define their specific interests to use those to help with the filtering and ranking of the resulting ranked list of citations being recommended. But, of course, it is much easier to use the user profile more on the filtering side of the recommendation approach since it is more a matter of just deleting rows which do not seem to correspond with interests. But, a combined approach should include the filtering too in order to be most useful.

Experimental Results

I first prepared all the background research and discussed many different approaches to building a better way to search and make recommendations for Semantic Medline. At the same time, I spent countless hours building the

technical infrastructure to provide access to Semantic Medline database and to be able to search it effectively and quickly in an environment outside of a huge clustered high performance system. This alone was a huge challenge and took many weeks and months to achieve. And, that didn't happen without a great deal of trial and error during this time frame. Then, with the research background and technical infrastructure developed, I set out to build an experimental environment which gets to the essence of the problem here for improving the Recommender capabilities for the Semantic Medline data base by advancing critical factors such as have been discussed. Novelty, diversity, and serendipity are all key goals for advancing the search capabilities. I have in fact realized many novel and interesting results after running the queries with the system now after many weeks of trials and experimentation.

Results of analysis of Semi-metric and Indirect associations

Proximity, distance, and similarity are the terms I have researched which measure the connections between concepts and predications as I have discussed in the preceding research. I have chosen from those approaches to be more in the distance category and I express the associations more by ranking they response by distance which as you may recall is basically the inverse of proximity or similarity of prefer that word. And, the distance is very similar to a page rank in many ways because they both do measure the probability of the occurrence in the database. I can use distance then as a measure for document ranking and equate that conceptually to page ranking in web searches. Really

very analogous concepts and fits well our Recommender Systems thesis here to help us support providing method recommender listings for medical researchers from Semantic Medline in this case.

Those connections can be direct or indirect connections. Or in other words, the direct edges in the graph database may not tell the whole story regarding a given concept or predication. I may also need to look beyond those to indirect connections through other objects to measure the total distance which I have just summed to get a final result in those cases and use that in our final ranking by distance. I have a method embedded in our code to do just this and it is very analogous to page ranking as have stated. But, the page ranking uses total distance and I can factor in the indirect connections as well. However, with such a huge database as Semantic Medline, we can quickly get beyond our capability with the technical infrastructure if we go too far in the indirect chain. So, we use measures like mean distance to cut it off and take all the ones whose distance is less than a mean value in the final results. And, this is all implemented with counters and code to make the accumulation of the results automatic for analysis. All this takes huge amount of time to code, text, and demonstrate. But, we have it in place and we can use it to get the results as posted in the Appendix for a few sample searches.

Comparison or Results

After many trials and adjustments to the technical framework to carry out the experimental approach, I prepared data tables for comparison of the results

for each Semantic Medline query. There are five such queries being used for the analysis. These range from a gene which was used in several examples shown already called BRCA1 to several diseases including the other examples which is diabetes. Each output table is labeled with the specific query for which the results pertain.

1. BRCA1
2. Metabolic Diseases
3. Malignant Neoplasms of Breast
4. Sporadic Breast Carcinoma
5. C11orf30

As stated previously, for validation testing to compare the results from the experimental platform with those from the MySQL database of Semantic Medline, I first ran queries in MySQL which created a table of the direct connected predications to the subject concept. Then, from the experimental platform, I ran queries against the Neo4J database to get results for direct and indirect connections and created tables with those resulting predications. Then, I was able to make sure all the predications found in the original MySQL tables were also in the new Neo4J output tables. And, we did find a 100% validation they were in the Neo4J query as expected. This test served to validate the data to the MySQL database.

And, as stated previously for performance testing, I prepared the best list of expected output predications possible with the experts available. I researched

the connections thoroughly to prepare the expert filtered list of predications which track to citations only using predications contained in Semantic Medline. Then, these were ranked ordered by the expert to create a list of predications which best matched the concepts. This was double checked with another expert to arrive at a consensus final list. Since all the queries are tested against the same final list, the relative difference is valid to show relative comparison between methods. Also, in order to collect the data quickly, I built code to compute the entries in the matrix automatically. This enables the analysis to be repeated with new queries much quicker with greater automation of the process.

So, the combined results from these tests are as follows:

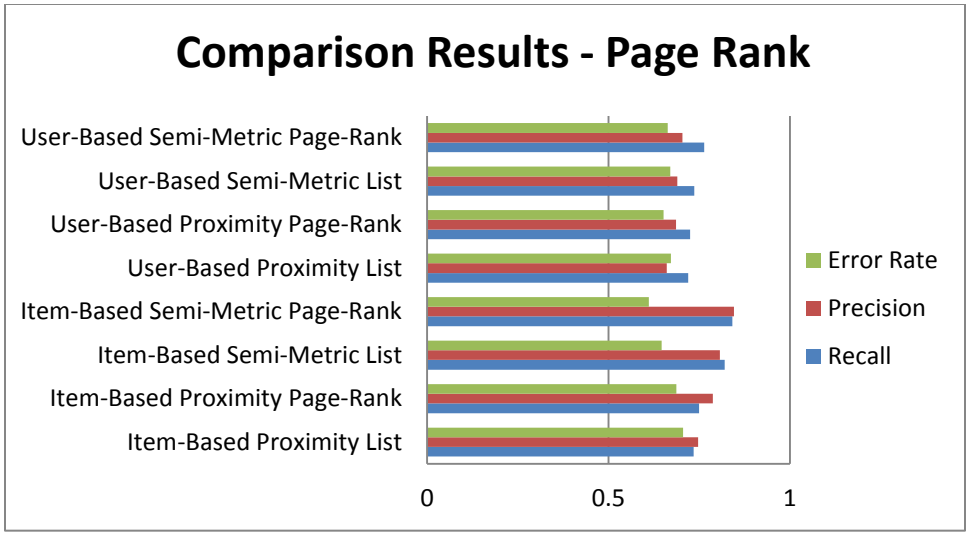


Figure 28 - Page Rank Comparison Results Bar Chart

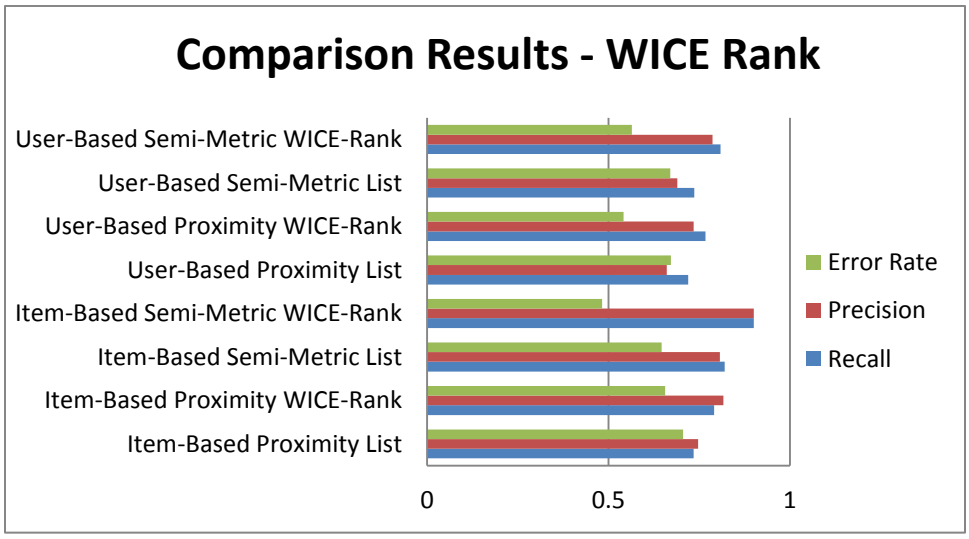


Figure 29 - WICE Rank Comparison Results Bar Chart

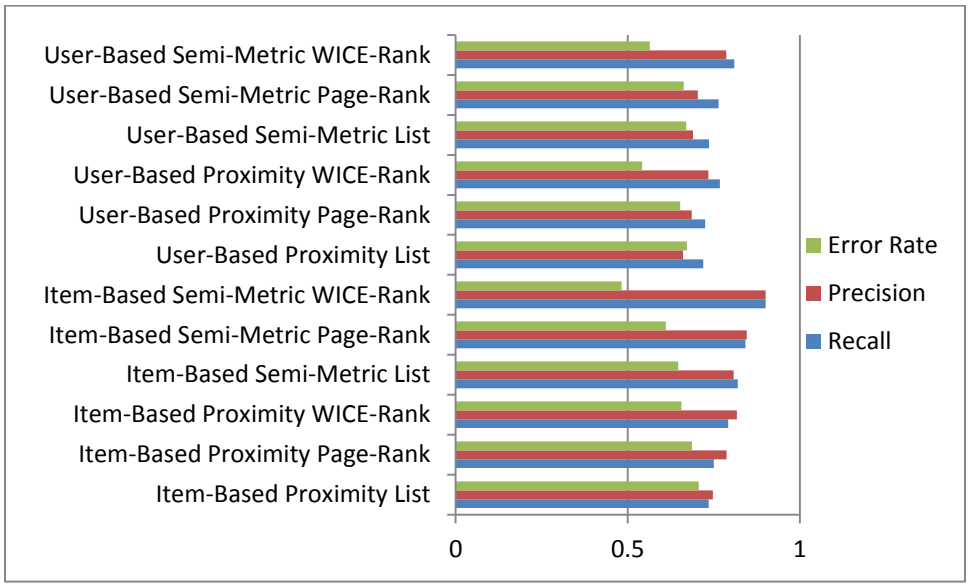


Figure 30 - Overall Comparison Results Bar Chart

Combined Results all Medical Queries – BOTH Rankings

In the final comparison chart it is the Item-Based Semi-Metric Ranked method which yields the highest recall, and precision with the lowest error rate overall. This is true whether the ranking approach was WICE or Page. WICE Ranking was actually better overall as can be seen very clearly. It is bit higher than Page Rank in each case. This probably makes sense because the User-based approaches are filtering by the User Profile and the reference list doesn't address the recommendations based on any particular user preference. The ranked methods are consistently better than the list methods. And, Semi-Metric is better than non-Semi-Metric because it finds novel new connections not found

without the Semi-metric additional links. So, generally, the results make sense intuitively.

And, overall, it is encouraging to see that the hit rate was very good with these approaches. Getting up to 90% precision is actually very good. The error rates are somewhat high but that figures too really. I am basically using our own reference list for comparison. I would expect a high error rate actually. An increase in the number of trials would lower the error rate. And, of course, the ranking methods do improve with time as more users do trials since the rank is stored and improves for subsequent iterations. Clustering with machine learning tools before the trials would have improved accuracy as well. But, with the resources available and the time required to achieve clustering while still maintaining an interactive user environment, clustering with such a large graph database was not practical in reality. Providing a basic clustering approach from the considerable research provided here would have been quite a lengthy process with the technical infra-structure available. And, such clustering needs to be accomplished as the data accumulates in Semantic Medline due to the increased processing power required to cluster after loaded to the database. So, I would advocate for providing an improved clustering approach as data is loaded. And, of course, pre-processing of the data is accomplished in a semantic manner now as discussed previously. I would advocate for providing an improved clustering approach as data is loaded. And, of course, pre-processing of the data is accomplished in a semantic manner now as discussed previously.

However, more clustering is called for as well using some of the methods discussed in this research. Just loading Neo4J from the .sql files took literally weeks to accomplish the database load. You can imagine what it would take to cluster this graph to a sub-graph level. I limited the graph environment to the predications alone to accomplish the Semantic Medline optimized algorithms. Further clustering would have greatly increased the load on the system to a level making it difficult to test.

The technical environment established was sufficient but of course, any time you are running with such large graphs, more is always better. And, that is also the case here. I was limited by run time all through this project. But, I was still able to get conclusive results with what the available environment. And, now, the Ruby approach could be easily implemented online since it is made for an internet environment using the architecture shown in previous diagrams. And, Python is good as a development environment because of its flexibility. Python was our development system where we optimized and proved the new combined algorithms were functioning well to achieve improved accuracy and novelty. Then, once perfected, the code was translated to the Ruby environment for implementation with visualization tools and for final testing.

Regarding some of the other important areas for evaluation with Recommender Systems such as Diversity, and Novelty discussed in the Appendix, I did find the indirect methods utilized here to be very useful in finding new associations which would not be picked up today with current Semantic

Medline search approaches because of the graph database and because of the algorithm's use of indirect association methods. I did not try to measure this but after looking at a number of novel indirect connections not directly tied to concept or predications, there were a many unique references provided which were in some cases with a shorter proximity distance than direct connections. I have attempted to quantify these to show a measure of the number of additional Novel citations found. And of course, there are many more with the Semi-Metric approaches. The numbers are actually found in the data collected. And, although the improvement is not huge, there are clearly more in the Semi-Metric as can be seen with the graphs provided. Clearly, our search algorithm did pick up a large number of important new connections which would not be found through other means. This too lends considerable credence to the methods tested here based on the research.

Building on these results, I have subsequently moved to implement features built on top of the edge ranking approach which begins to incorporate link prediction such as found in the "Friends Finder" capability in Facebook. It runs on the existing graph and uses existing connections to train a model for predicting new links similar to existing links especially in cases where I have indirect connections between nodes without defined connections. Since Semantic Medline is a very large graph, it is important to use scalable approaches like edge link prediction algorithms of which an adoption of PageRank is one of the methods. Since I have found some limited success here

with edge and page rank, it makes sense to extend these for link prediction. Google searches have exposed a number of Python references which could be used for experimental exploration. Some of these are being implemented and new experiments being created beyond those already conducted here.

Throughout the process of experimentation, algorithm iteration, it is important to list some of the many trade-offs encountered in the process. These are listed as follows:

- Size of the database and time to run processes increase quickly when going beyond 1st degree of separation from a given node. Therefore, steps need to be taken to limit some searches while still finding useful candidates for final lists.
- New users need results quickly at time of use. With a large graph database, this becomes a major design concern. Every process needs to be evaluated for response time before development.
- Longer term users can build profiles and accounts for use later with results from longer term threads. Design of a good solution for Link Prediction must assume processing in the background to achieve needs here with results on next login.
- The Semantic Medline database is preprocessed to use common concepts and connections. With such a large database, additional preprocessing is very time consuming in any environment.

- As with all such Information processing projects, the scale of the infra-structure has a major impact on performance. Although very substantial, the hardware available to me for fast access and iteration of trials was somewhat limited.
- Because of somewhat limited infra-structure, every process design was optimized to requirements. This is a positive result for our solution in terms of implementation for a production solution. The design is highly optimized.
- No good solution was discarded based on performance in the test system. Any solution discarded was based primarily on suitability to Semantic Medline.

By trading-off constraints and iterating solutions, I did find optimized algorithms which do improve the accuracy and novelty of searches in Semantic Medline based on test results. However, because the algorithms utilize ranking approaches with iterate and improve over time, a fully implemented system which contains these innovative algorithms would adapt to the usage which was encountered by groups of users making use of the recommendations provided. So, the result becomes more important from the perspective of being a self-adjusting system which dynamically adjusts to user interests over time. And, although it is important to have experimental results which show improvement in recommendation accuracy, this accuracy will improve with usage which reduces the important of experimental validation.

Summary of Results

1. **Validation testing** proved that all of the optional solutions developed fully include resulting citations of the existing Semantic Medline MySQL system as a baseline.
2. **Performance testing found:**

a. Rank algorithms perform better than options without Rank.

b. WICE rank performs better than Page Rank.

c. Item-based queries perform better than user-based queries.

d. Semi-metric based queries performed better than the Proximity queries overall.

e. Item-based Semi-Metric with WICE rank exceeds all the other options across samples.

f. Rank algorithms improve accuracy with usage as rank values iterate to conform to user interests.

g. All of the performance tests provided responses within reasonable time frames normally associated with interactive user expectations.

h. All results were validated when considering inherent testing errors. Since ranking algorithms adapt to the user interests, inherent error also reduces with usage as the ranks conform to fit the user interests.

3. **Novel recommendations** were common with the Semi-Metric options.

Conclusions

This research has found the best methods for filling many of the gaps found with document network systems available today for medical research using Semantic Medline. A summary of the major advances from this research is as follows:

1. A Graph DN implementation with algorithms optimized for the DN does improve recommendation accuracy and novelty by casting a wider selection set and ranking in order of predicted preference.
2. Big Data migration from SQL to Neo4J graph database provided many search benefits efficiently. With big data, multiple joins will not function efficiently, if at all.
3. Indirect Semi-Metric recommender methods do yield novel and accurate results.
4. Proximity and similarity measures do yield novel relationships with greater accuracy.
5. Recommender systems methods do greatly improved the accuracy for DN.
6. Ranking algorithms do improve recommendation accuracy for a DN. And, the accuracy they bring improves with user interaction.
7. Link predictions algorithms using ranking do predict links with improved results.

8. User profiles do improve results for a user when measured from the user perspective.
9. Gaps identified initially are filled by this unique solution adapted to DN.
10. A well formulated blend of the combination of these tools adapted to the specific Document Network does improve the overall accuracy of predicting the 'rating' or 'preference' that a user would give to an item of information from the document network over conventional approaches without these improvements.
11. Methodology used here is generalizable to make use of Recommender Systems and graph database tools to improve recommendation accuracy.

For this case study for Semantic Medline, with the technical infrastructure I had available, I found an innovative approach to providing much improved research tools for researchers to find pertinent medical research citations for specific research topics. I looked at all the important aspects of this kind of Recommender System requirements, I reviewed literature for ideas of how to proceed, and I built a system which loaded the database and made it available in a graph database format to run experimental approaches to find the best solution for Semantic Medline, the largest and most important document network for medical researchers available. The processes used in this research include all the important aspects necessary for such a system to fully cover the

requirements. And, I found ways to innovate our own approaches for the solutions based on past research. Working with the trade-offs in my system environment, found a reasonable approach to balance the methods employed. This research can be used to actually build a platform which could be used online for the medical research community using the approaches innovated here. And, this can add to the value of the database as it exists today.

This Dissertation is meant to be an initial research in the pursuit of better approaches for making recommendations for documents in document networks. I have reviewed a number of approaches and have focused on those with the best possibilities. Within those versions of the optimized DN algorithms, there are many additional sub-options which can be considered as well considering the many ways algorithms can be developed for Page Ranking and Edge Ranking for example. And, there are methods which can be employed to improve what has been started here. I would encourage additional research to further explore those possibilities and grow the knowledge available for this subject area.

The principal achievement I have in this research is bringing forward several optimized versions of the DN algorithms to address the research objectives considered here. And, working through those versions to find the best practical approaches and conducting experimental research on the most promising algorithmic approaches. Clearly, a well-orchestrated combination of Ranking algorithms and Semi-Metric item based approaches are very capable in this regard for very large databases which have been pre-processed like the

Semantic Medline database. Just to accomplish our objectives to this level with big data has required many months to identify all the relevant research, build an experimental platform for testing many of the optional approaches, learn the best technology to fit the problem with Neo4J, Python, and Ruby on Rails Framework, and coming to an understanding of best methods to solve the problems as they presented and work out the many trade-offs.

And, beyond these achievements, I have started here a major new initiative to explore methods for link prediction based on these results which have already appeared to offer considerable potential with trials already conducted. This moves the capabilities in the direction of the Facebook capability for "Friends Finder" which is a powerful predictive technique for identifying possible new connections based on training a graph model using similarity measures to predict new connections and then, giving the user a chance to look into those possible predictions to ascertain relevance to their research focus. This is an important add-on capability to those already developed and tested here which can be implemented to surpass presently proven approaches through this research.

Also, I have reviewed many different approaches to the general objective of using context to support the processes inherent to Recommender Systems. Again, we are considering Recommender Systems from a very broad perspective which includes information systems which support search, decision making, social networks, and commercial advertising systems. Nearly all of these

approaches do offer evidence that the use of semantic technologies to include user context would be beneficial to the process and make the results more accurate, scalable, and provide an overall improvement if it can be done through more implicit or inferred methods.

However, the case studied in this research has not actually provided an approach for user context which fully achieves the objective for more generalized model. The approach used here is oriented to building a user profile to help filter documents not of interest to the user which is a limited approach. Clearly, this area is ripe for much more innovation and considerable additional research. And, the goal of creating a user model which includes context that could be utilized by generalized content for recommendations is clearly where the additional research needs to go. There must be considerably more research going on in these areas but it is not found in recently published articles at least so far. And, clearly, the benefits of advancing such research would be considerable for the economic goals of entities which may pursue the research to implement improved recommender systems.

If we are pursuing a semantic approach to the inclusion of context in the process, then how can this be done implicitly or inferred behind the scenes without asking the user questions on an explicit basis to populate a user profile? For the semantic approach to function well, the user model needs to be an ontology which can populate itself because a manual approach would be too time

consuming to be effective. [29] So, the research needs to continue to experiment with semantic user context models which are self-populating.

Additional future work which would be beneficial are as follows:

- Continue to implement **link prediction** algorithms as threads in background to recommend new links to users. Initial tests prove to be very promising using the ranking algorithms perfected in this dissertation.
- Test the code in an **AWS** environment so can make available for broader testing and refining the optimization of the system with more trials.
- Perform **trials with more users** while improving the user experience and expanding their profiles.
- **Broaden the number of test cases** and get more expert validation of the reference evaluation data sets.
- **Scale the technical platform** to enterprise level capability so can implement full user functionality.
- Build in connection to groups of users who share similar research interests to expand on the **"Shared" Interest graph** concept.
- Further explore the use of **SPARQL** to provide additional features.
- Extend the graph database to include predications from **the full text of the documents** referenced with Semantic Medline

.

This dissertation has focused on improved recommender accuracy and precision for Document Networks using Semantic Medline as the case study within the Dissertation. And, greatly improved results were obtained. These additional activities would continue the progress for Recommender Systems in Document Networks. My novel optimized algorithms can serve as the basis for these additional research activities.

APPENDIX

Background and Papers for Technological Basis

A broader background elaboration on Recommender Systems and how they are Evaluated for specific cases is provided here in order to include broaden the Dissertation to other cases beyond DN and Semantic Medline which are a unique case. I also include information on algorithmic research which could be useful for other cases as well for the same purpose. This information was useful to narrow down the best approaches for the case study presented. But, since it did not directly apply to the case study, it was put here in the Appendix since it was integral to the Dissertation process.

Further along in the Appendix, I have included the actual data collected for the experimental research to reference how the tests were conducted in detail. The results discussed in the body of the Dissertation makes use of the data provided here.

Recommender Systems –

The general subject for discussion addressed in this paper is Recommender Systems. The term ‘Recommender System’ is a broad area within automated information systems which refers to systems which provide a service

based on an explicit, implicit, or inferred knowledge regarding the user of the service to offer recommendations which help the user to obtain the information desired. These recommendations can be offered by the Recommender System when searching for information through a search engine, interacting with an e-Commerce site to find products and services to fulfill the needs of the user, through interacting with other users in a social networking environment, or generally browsing sites to obtain information pertinent to the interests and decision making needs of the user. The marketing arms of enterprises wish to offer products to users which can meet the needs and preferences of the user while they interact with information systems. Users wish to efficiently utilize information systems to find the information and knowledge they need. So, Recommender Systems can provide a very useful service to all involved. And, the context of those interactions has a very important role to play in the outcome realized.

To understand how we need to focus our research into Recommender Systems for Document Networks, it is a good idea to explore the types of Recommender Systems in order to determine the characteristics of importance for this research. There are a number of approaches for providing Recommender Systems which divide the universe into these types: [79]

Traditional Methods [79]

1. **<u>Collaborative Filtering</u> based on making recommendations based on historical usage behavior from:**

a. **User based:** Find similar users and recommend what they liked.

Users have list of m Users and n Items to measure similarity between users. Then, select a subset of neighbors similar to the user doing search. [79]

b. **Item based:** Each user has a list of items with both explicit (rating score) and implicit opinions (purchase history) with a method to predict a rating by user.

c. **Model based:** Uses the entire user-item database to generate predictions and uses statistical techniques to find the neighbors (nearest neighbor). First a model of the user is prepared using clustering, rule-based approaches, classification and other computational methods.

- Clustering: Cluster is assigned preferences based on users in the cluster. Users in a given cluster receive recommendations for the cluster

- Locality-sensitive Hashing: grouping similar items in dimensional spaces. Main application is nearest-neighbor algorithms using hashing and high performance.

- Association Rules: Past purchases are transformed to relationships of common purchases. Then, these rules used to make recommendations.

- Classifiers: Models trained using positive and negative training examples which can include vector item features, user preferences, and relations between inputs.

Basic Steps for Collaborative Filtering process: [79]

- Identify set of ratings for user making requests.
- Identify set of users most similar to user using similarity function
- Identify the products these users like.
- Generate a prediction based on these factors for each item
- Based on predicted rating, commend top candidates

2. **Content** based which makes recommendations from item features to match the profile of the request with following elements: (Document Network recommender systems such as Semantic Medline fall into this type of Recommender System) [79]
   a. Based on content of items and not on user opinions or prior interactions
   b. Use machine learning algorithms to induce a model of user preferences
   c. Based on similar items a user liked in the past.
   d. Content based only by analyzing content of the items requested.
   e. Content is usually described with keywords where preferences for an item are achieved by analyzing the content of previous items and by keyword analysis.
   f. The importance of a keyword is determined by a weighting measure such as computing the distance between documents using that to recommend closest items in the list.

g. Other techniques are feasible such as using classifiers and machine learning techniques such as clustering, decision trees, and Artificial Neural Networks.

Novel Methods [79]

3. **Rank Learning** based which treats the recommendation as a ranking problem. [79]
    a. Most recommendations are provided from a sorted list
    b. Recommendations are understood as ranking problem typically using machine learning tools.
    c. The item popularity is a typical baseline
    d. Personalized ratings can be a secondary input
    e. Many other features can be added to the ranking criteria.
    f. Resulting order of items typically computed as a numerical score.
    g. Can be treated as a standard supervised machine learning classification problem which is somewhat difficult to optimize.
4. **Context-Aware** based recommendations are generally implemented with these types of architecture: Context Pre-Filtering, Contextual Post-Filtering, Contextual Modelling. Combinations of these types are possible.
    a. **Pre-Filtering:** uses context to select the most relevant data for generation of recommendations.
    b. **Context Over-Specification:** Exact context may be too narrow for the recommendation. So, the important aspect of the context is prioritized.

c. **Pre-Filter Generation:** There are a variety of approaches but they can "roll-up" to higher level concepts into context hierarchies.
d. Ignore context in the data selection and modeling but do filtering and re-ranking based on context information.
e. **Post-Filtering:** treats context as another constraint and have many approaches.
f. Can also use context directly in the modeling or learning phase and even be added to the dimensions of the data.
g. **Tensor Factorization:** A pre-filtering approach which computes recommendations using only the ratings in the same context as the target and splits items where there are significant differences in their rating under different context situations.
h. **Factorization Machines**: Factorization is combined with linear regression and requires new learning algorithms where the input is treated as real-valued feature vectors. [79]

5. **Deep Learning** methodology to perform predictions from ANN training sets using GPU's with CUDA code and AWS for big data access.
6. **Similarity** based on different dimensions which can refer to metadata/tags, user behavior, or rating behavior such as SimRank. A score is derived for the various dimensions which are combined into a weighting approach using regression.

7. **User Demographic** based recommendations on user profile features by categorizing user based attributes and is based on demographic classes. The demographic groups can have research associated with the group and form techniques for user-to-user correlations. Demographic features are requested from the users but can also be induced through classification. Prediction can use learning tools like nearest neighbor or naïve classifier.
8. **Social Network** based recommendations on user social network and based on social proximity of the user with the assumption of trust in judgement as a central idea. Trust is based on past interactions and is used to describe similarity of opinion with a goal where the source and sink nodes have a value on trust. Trust can help with giving more weight to users, collaborative filtering, and for sorting.
9. **Hybrid** which consist of a combination of any of the above types. Content-based with classifiers, collaborative using correlation, and collaboration with content based user profiles are a few examples. The following hybridization methods:
    a. **Weighted** – Combine results of using different techniques into one list where the relative value of the different techniques is fairly uniform across the option

b. **Switching** – Users criteria to switch between techniques when one method has lower confidence in certain cases than other techniques. The biggest problem with it is to identify valid switching criteria.

c. **Mixed –** Recommendations from more than one technique presented together.

d. **Feature Combination –** treats collaborative information as additional features for a given example set and can also treat content features as different dimensions for a collaborative setting.

e. **Cascade –** Use one technique first to produce a coarse ranking and a second to refine the recommendation. But, this requires ordering of the techniques.

f. **Feature Augmentation –** Produce a rating or classification of an item and this rating is incorporated into the next technique and is very similar to Feature Combination.

Pros and Cons of the different methods-

1. **Collaborative Filtering** – [79]

Pros: Requires least knowledge to predict with object, easy to measure information and often yields good results. Clustering techniques can work with aggregated data, can be used to shrink selection set to neighbors, and can be used to capture latent similarities between users. Association rules are fast to implement and execute, require little storage space, and very successful in broad applications. Classifiers are versatile, and can be combined with other methods.

Cons: Requires a lot of reliable data which can be quite dynamic, similarity computation is inefficient, requires more standardized items, and assumes prior behavior is a good predictor. Cold start recommendations with new users difficult. With clustering, recommendations may be less relevant to the members of cluster. With association rules, they are not suitable when preferences change rapidly and can lead to wrong recommendations. Classifiers require a training set

2. **Content Based –** [79]

Pros: Don't need user data so cold starts not a problem. Can address unique interests of requester and find new, novel items not based on popularity. Can explain description as to why recommended.

Cons: Requires content is pre-processed to expose meaningful features. Can be difficult in some cases to provide comparison between non-quantitative information and difficult to exploit quality judgements.

Recommendation processes are a general data mining problem with all the elements of such problems to include the following: [79]

1. **Data preparation –** Feature Selection, Dimension Reduction, Normalization, Data Categorization. Document Networks actually make good use of data preparation to capture relationships and to computer associations since the documents don't change over time. Only new documents are added. When added, their characteristics are computed when included in the graph database.

2. **Data Mining –** Clustering, Rule Generation, Classification
3. **Post-processing -** Filtering, Pattern Recognition, Visualization

The Recommender System we are working with in Semantic Medline is an Item Content based extracting features on items and comparing similarities. This is different than the user-based recommender system because it does not include user data. A node is the items in the DN and the relationships are the features. In Medline, the items are the medical concepts. And, the features are the relationship as extracted using natural language algorithms. We are recommending citations based on existing relationships between documents. The Semantic Medline database once analyzed included many self-directed relationships which we did not want to include in the approach. So, filtered out these isolated nodes since they were not relevant to the type of approach we intend to study here.

Evaluation of Recommender System Performance

For this project, we need to evaluate the experimental results in a manner which relates to important characteristics of the Recommender Systems studied. In order to accomplish this evaluation of results, we need a framework for the evaluation process based on the computational methods to be evaluated. So, we need to layout important evaluation approaches and criteria for this purpose. This section presents the evaluation approaches available for Recommender Systems along with the considerations for the available approaches so we can utilize this

information to form a basis for the evaluation framework for the present Dissertation.

There are a number of ways to perform property-directed evaluation of recommender systems. For each type, there are a number of properties that can be relevant for system success with rankings of how a candidate system can perform with respect to the properties. Experiments can be conducted for the various properties to perform the evaluation and the evaluation metrics associated with each. [67]

For an evaluation method, we need the typical scientific method approach which starts with a hypothesis and control variables. An example of a hypothesis might be algorithm A predicts user ratings better than algorithm B. The generalization scope of the experiment is a measure of how a result can be used in other scenarios. For the experiment, one needs to have data valid for the test scenario and a way to simulate user behavior to test the hypothesis. Sampling of test sets can be used to reduce the cost of testing. And, time it takes to do the tests can be ignored as not important to the evaluation method. But, the test sequence is an important test simulation variable because it needs to simulate the recommendation process to a large extent.

A common protocol for evaluation would use a fixed number of known items and a fixed number of unknown items in order to diagnose algorithms and see which work best. But, with a fixed number then we will not know what happens precisely when more or fewer items are used in the recommendation

algorithm which then is an experimental bias from the evaluation approach used. So, this is one of the decisions need to be made to set the evaluation methodology along with many others such as user modeling of test sequences.

User modeling is very difficult and it can as well lead to evaluation biases and less than optimal recommendation performance. More complex models require care in generalization of the results since more difficult to verify the models. Less complex models can mean important variables are not been analyzed with respect to the recommendations provided. So, the level of user model complexity becomes an important aspect of the evaluation method design.

When conducting evaluation with **user studies** as one form of evaluation which is important for recommender systems, subjects are asked to perform a number of tasks to record behavior and to quantify the measurements such as percentage of the task completed. Then, the quantitative information is collected along with user experience information on the tasks provided. These kinds of user studies are a central tool for evaluation. The advantage is providing a capability to test the user behavior with the recommendation system and to collect qualitative information which is often very important for the interpretation of the quantitative results. However, these user studies can be expensive and time consuming. Therefore, normally, a very small set of test subjects are used for such user studies in practice where the sample of users is drawn to be representative of the global user population to extent possible. This then

balances the costs with the ability to general the results to the general population if the sampling is conducted properly. [67]

Questionnaires are often used prior, during and after user studies to ask about the experiences. While these surveys can provide good information, they can also be misleading if not administered well. It is important to ask neutral questions when administered to reduce survey bias and to improve generalization of results. These surveys combined with **online testing** can provide best results overall because with this approach, the evaluation can obtain a more diverse set of results gathered from users on different systems to get a better cross section of usage built into the evaluation results. Offline studies are good to collect information and tweak the online approach. Then, once online, the pre-testing of the evaluation process can lead to lower risk of user dissatisfaction. [67]

Drawing Reliable Conclusions

In order to assure reliable conclusions, it is important to perform significance testing on the results of the experiments. A standard tool is the significance level or p-value which is the probability that the results found are due to chance. Typically, one rejects the null hypothesis that the algorithm A is no better than algorithm B if the p-value is above some threshold value. In other words, if the probability that the observed ranking is achieved by chance exceeds the threshold, then the experimental results are not considered to be significant. Usually, $p = 0.05$ is chosen for the threshold which would indicate a less than

95% confidence level. Higher levels of confidence can be used when the costs of making wrong decisions are high. [67]

In order to actually run a significance test, it is first required to have several independent experimental runs. As discussed leading to this point, test data needs to have been generated carefully following a certain protocol and test users must be drawn independently from the population. And, the best approach is to compare algorithms on a per-user case basis as opposed to per-item since it is unlikely the items are independent. So, then the process to obtain the significance becomes to count the number of users where algorithm A outperforms algorithm B and the number where B outperforms A. The significance level is the probability that A is not truly better than B. Another approach is to look at the average difference between performance scores on algorithms A and B, normalize with the standard deviations of the score difference to get the resulting better algorithm. To improve the overall significance, larger test sets are beneficial but of course, there are trades-offs there with time and expense. This is all more complex if using a larger number of algorithms. So, it is often good idea to compare two at a time to get down to the two best algorithms for the final trade-off. [67]

Recommender System Properties

There are a number of properties to consider in order to select a given recommendation approach. Each recommender application has different needs. So, the one selected depends very much on how the application matches up to

the properties. And, trading-off the properties is a necessary step as well in order to select the best approach since they properties can conflict to some extent. The properties have parameters which can be adjusted to adjust the recommender response as well. So, these need to be identified in the process of trading-off properties to select the best approach too. Once the properties are well understood along with the parameters to make adjustments, then a selection is a result of the analysis. So, we now review the properties generally available for recommender systems. [67]

User Preference

We discussed user studies which can lead to a good idea of user preferences. In the user studies, we found a wide range of biases which can result. From the user studies, we can select the system with the largest number of votes. But, this scheme assumes all users are created equal as the votes are counted equally. This may not be true in practice. For example, an e-commerce site may prefer the opinions of users who buy multiple items as opposed to those who buy one. Therefore, we need to weight the votes by the important of each voting users when this is applicable. But, assigning appropriate user weights is not exactly an easy process either in practice. It is also possible that the difference in opinion between the systems is small too. In this case, even though A is slightly more preferred, B may be a better answer. So, to manage this we need to have votes be non-binary and result in a score of some sort which is appropriate for the system being tested. And, it is also important to know why a

user has certain preferences and break the reasons down to components for evaluation. So, selection of best system is not exactly a straight forward process and it does require some agility to get the right answer depending on the application. [67]

Prediction Accuracy

Prediction accuracy is the principal way recommender systems are discussed in the literature. The basic assumption for the evaluation of prediction accuracy is that the user prefers more accurate predictions than less. Accuracy is generally independent of the user interface which opens up the option of using off-line experiments. The measurement of prediction accuracy in a user study depends on the measurement of the accuracy of a specific recommendation for the specific user. This is a different concept than the prediction of a user behavior without a recommendation. This leads to a discussion of the three broader areas of predication accuracy as follows:

1. **Measuring Ratings Prediction Accuracy –** In some applications, we are predicting the rating of a user for a given item. In such cases, we want to measure the accuracy of the system's predicated ratings. Root Mean Squared Error (RMSE) is one of the more popular metrics for accuracy of ratings. Then, there are the normalized and average RMSE to adjust for unbalanced test sets.
2. **Measuring Usage Prediction –** If not predicting ratings, the experiment may be trying to predict items the user may prefer. In this case, the

accuracy is dependent on whether the user would add the items to their selection queue. From a sample off-line evaluation, we might have the data set of items the user has used. So, to test the user, you can hide the prior selections and ask the recommender to predict the set they would select. Then, measure the difference. So, there are 4 possibilities which are recommended, not recommended, used, and not used. Then, count the items in each cell of the matrix to compute the Precision at N for a fixed length N of items. But, over a range of list lengths, we can compute curves comparing precision to recall known as Receiver Operating Characteristics or ROC. The curves basically represent the proportion of preferred items actually recommended. Precision-recall curves emphasize the proportion of recommended items that are preferred while the ROC curves emphasize the proportion which are not recommended. Given two algorithms, we compute the curves to see if one curve completely dominates the other, if they intersect, or just what we get as an outcome and act accordingly on the results. And, then we have all this analysis with potentially multiple users which complicates it even further but generally gives the evaluator a set of curves in order to trade-off the systems based on a set of parameters as discussed earlier. [67]

3. **Ranking Measures**

Recommender systems often present a list of recommendations where the objective is ordering the items according to the user's preferences. This is the

predominate form of recommendation. So, ranking of the list is the important aspect after the list has been selected by the system. To evaluate the approaches, we can measure how close the system comes to the best order for a user or we can attempt to measure the utility of the system's ranking for a user.

a. **Reference Ranking**

To evaluate the ranking, we first need a reference ranking which is typically in decreasing order of preference to show the most preferred first in the list. When only usage data is available, the reference ranking may be a list of items actually used by the user with those used being listed before those not used. This is only valid actually if we know the user was aware of the unused items. So, logs can help us to determine a reference ranking to construct a reference list and order the list. Then, the ranking provided by the system can be compared to the reference listing to get a measure called Normalized Distance-based Performance Measure (NDPM) to help measure the system versus the reference list where the score is perfect if the system provides a list the same as the reference list. The worst score is 1 where the actual list completely contradicts the reference list. [67]

b. **Utility-Based Ranking**

Another way to provide an evaluation measure of ranking is to assume that the utility of a list of recommendations is additive given the sum of the utilities of the individual recommendations. Here the user reviews the list and the probability that a specific position is observed by the user depends on the

position in the list only and not the items in the list. An R-score metric assumes the value of the recommendation decline down the ranked list yields a specific score for a small number of items in a ranked list. The resulting per-user scores are aggregated to find the best possible ranking for a given users. For larger lists, the user needs to assess a larger portion of the list and a measure called Normalized Cumulative Discounted Gain (NDCG) is used to measure item positions through an inverse logarithmic function. Both cases contain a utility function to assign values for each items which can be replaced with a function appropriate for the given system design. [67]

c. **Combined ranking method evaluated online**

When evaluation of the method is online, the system will provide a ranked list and the user can select those which are of interest to them into 3 parts for those (1) of interest, (2) of no interest, and (3) unknown interest. Then, the evaluation can use an appropriate reference list ranking metric to score the list for those of interest and for those not of interest using the reference list. Also, the utility ranking can be used to measure by item position in the list. So, a combination of methods can be used here. [67]

4. Coverage

Prediction accuracy grows with the amount of data available to the system. But, the algorithms may only provide quality recommendations for a portion of the items within the larger data sets. So, the term coverage refers to properties based on the data processed and leads to properties as follows:

a. **Item Space Coverage**

Refers to the proportion of items the system can recommend and is measured by the percentage of all items that can be recommended. It can be computed directly from the given algorithm and input dataset. In some cases it is useful to weight the items for coverage in the dataset. Another measure is sales diversity which is how unequally different items are chosen by users when a specific recommender is used. Another measure is the Shannon Entropy which is zero when a single item is always chosen and log n when n items are chosen.

b. **User Space Coverage**

Coverage is also the proportion of users or interactions for which the system can recommend items. It can be measured by the richness of the user profile required to make a recommendation. Recommenders should be evaluated on the basis of their tradeoff between coverage and accuracy. [67]

c. **Cold Start**

Another coverage issue is the cold start problem when new items and new users are the subject of the recommender system. For cold start items, there is a measure of the threshold to decide on a set of cold start items where the time the items exist in the system and the amount of data gathered for them is important to measure their cold start property. There can be a tradeoff between ability to handle cold start and hot item accuracy to consider with the given method employed where the system is credited more for predicting cold items and less for hot items that are predicted. [67]

5. **Confidence**

Confidence is the system's trust in its recommendations or predictions. The system should improve their accuracy as the amount of data for items grows and as the predicated property data grows. One of the most common measurements of confidence is the probability that the predicted value is true or the interval around the predicted value where 95% of the true values exist. Given two recommenders that perform with similar results on other relevant properties such as prediction accuracy, it may be desirable to select the one with valid confidence estimates. Another use of confidence ranges is when the recommended items are filtered for a predicted confidence below a threshold value. Then, the experiment is designed around the filtering procedure to compare accuracy after removing low confidence items. So, confidence comparisons can be an important way to compare algorithms where all other properties are similar. [67]

6. **Trust**

Here we are referring to the user's trust in the recommendations. For user trust, a survey can be provided to the user to ask about the level of trust from the experiments by associating the number of recommendations provided with the trust in those recommendations. And, return rates to use the recommender can also be used to determine trust by the users although return rate may be difficult to separate from other factors for the evaluation conclusions. [67]

7. **Novelty**

Novelty is a measure of the recommended items not known to the user. So, for this we can filter out the items already rated or used by the user. Again, surveys can be useful to measure this property too and we can gain some understanding with offline studies. To do this, split the database according to timeframe and hide those user ratings which occurred over a specific time frame. Then, the system is rewarded for each item recommended and rated after the split time and punished for each item recommended but rated prior to the split time. This will provide a good measure of novelty. Another method to evaluate novelty uses the assumption that popular items are less likely to be novel. So, novelty can be taken into account by using accuracy metric where the system does not get the same credit for correctly prediction popular items as non-popular ones. Also, we can evaluate the amount of new information in a recommendation together with the relevance of the recommended items. When item ratings are available, we multiply the hidden rating by some measurement of the recommended item to produce a novelty score. Novelty is an important measure and it is important to maximize. [67]

8. **Serendipity**

Serendipity is a measure of how surprising the successful recommendations have resulted from the system. For example, recommending a new movie with a favorite actor is novel but not surprising. But, surprising may also mean inaccurate. So, serendipity needs to be balanced with accuracy. Put another way, serendipity is the amount of relevant recommendation which is new

to the user. So, to measure serendipity, we can measure the distance between items based on content. Then, we score those as successful by a distance from a set of previously rated items or from the user profile in a content-based recommender. We reward the system for successful recommendations that are far from the user profile. Serendipity can also be consider the deviation from the natural prediction providing higher serendipity scores to successful recommendations that the prediction engine would deem unlikely items. It is evaluated by asking users to mark the recommendations that are unexpected. Then, we see if the user used these recommendations which would make them unexpected and useful or serendipitous. [67]

9. **Diversity**

Diversity is the opposite of similarity which suggests that similar items may not be as useful because they may not explore the range of likely items for the recommendation. To measure diversity, often item content lists are used to first find the item to item similarity. Then, diversity is measured from the list based on distance between the pairs of items or by measuring the value of adding each item to the recommendation list as the new diversity compared to those already in the list. The item-item similarity is different from the similarity measure of the recommendation list. Diversity may come at expense of accuracy. So, diversity must be balanced with accuracy. [67]

10. **Utility**

We can define various utility functions for the recommender to optimize. For some recommenders, measuring utility may be more important than accuracy. And, other properties like diversity or serendipity can be viewed as different types of utility functions. Utility is defined as the value the user gains with the recommender. When users rate items, it is possible to use the ratings as a measure of utility with positive value assigned for successful items to add to the utility function, and negative for unsuccessful ones to subtract from the utility. If utility or value is the profit returned from a web site, then this is one measure which can be tracked with the financial outcomes of the system. Other kinds of utility are more ambiguous but can also be measured with ratings. [67]

11. Risk

In some cases, the recommendation may be associated with a risk. The way to evaluate risk sensitive systems is by considering the utility variance and not just the expected utility. Using a parameter to measure risk, when it is positive the approach prefers risk-seeking recommenders, and when negative it is risk-averse. [67]

12. Robustness

Robustness is the stability of the recommendation when there is fake information utilized which may be inserted on purpose to influence the recommendation. Influencing the system to change its rating on an item may be profitable. These influencing events are considered to be attacks on the recommender accuracy. So, the attack protocol needs to be related to the

sensitivity of the system based on the protocol being used. We cannot really prevent attacks but we can estimate the cost of influencing the recommender measured based on the amount of injected information. Another measure of robustness is the stability under extreme conditions like high volume request which is related to the system infrastructure. [67]

13. Privacy

Users prefer their preferences to remain private in order to prevent external systems to learn about users. So, the recommender may not expose private information to the public. Since privacy is never perfect, it is appropriate to define different levels of privacy such as a k-identity and compare algorithms sensitivity to privacy leaks under varying levels of overall privacy. Also, as with other properties, privacy has a tradeoff with accuracy. Changes to the algorithm can be evaluated for accuracy when a privacy modification is made and see how it impacts the performance. [67]

14. Ability to Adapt

Recommenders may function in an environment with rapidly changing data where trends may shift quickly as in typical e-commerce situations. With changes, it is a bit like the cold start problem where the recommender was trained for certain information to achieve expected results. With rapidly changing environments, the properties can change as well. So, ability to adapt is the ability of the system to remain within performance parameter when the environment changes. Another type of ability to adapt would be when user preferences are

changing at the same time where it would not be expected to have the recommendation remain fixed. So, we can evaluate ability to adapt by measuring the difference between the recommendation lists before and after the changes. The Gini index and the Shannon entropy can be used to measure the variability of the recommendations made to get some measure of ability to adapt. [67]

15. Scalability

For large collections of items, the recommender design needs to be able to scale up to real world situations with many items and many users. So, one approach is to measure the computational complexity in terms of time and space requirements along with the consumption of system resources over the larger data sets. Scalability is typically measured by experimenting with growing data sets and showing how the speed and resource consumption behave as the task scales up. It is important to measure the compromises that scalability can cause with accuracy and other properties. These measures will provide important information on the potential performance of the recommender going forward as load increases. Since recommender is expected to provide rapid responses, the measure of how fast it provides recommendations needs to be included in this analysis. [67]

So, how recommenders are evaluated and the properties which pertain to the success are important to understand when improving algorithms for certain tasks. Important considerations have been addressed which all pertain to the improvements studied in this research on Document Networks and Semantic

Medline. Each property can have an experimental approach and measurement to be considered in the final analysis. And, properties can be conflicting. So, the balance of the various properties needs to be considered along with the specific goals of the recommender application under analysis.

Useful Research Not Used in Case Study

This section is important because I am providing research which seemed to have a great deal of promise for the case study solution, but it was not used for a variety of reasons. It is important in the broader context of this Dissertation, however. Because many of these research findings could be more important for other cases. So, instead of dropping all of them from the Dissertation research paper, I have added them here to provide more information for the broader Dissertation topic.

Similarity Measure using Confidence Factor

Another way to look at the association between documents in a DN is to consider the similar case of web pages to determine the closeness of such pages. For this proximity between web pages, the value of the direct confidence is considered which computes a value of a direct confidence function which denotes the belief that the page dj may be recommended to a user while watching page di. In other words, the direct confidence factor is the conditional probability that a session that contains di will also contain dj.

Equation 5 - Confidence Factor [64]

$$con(d_i \rightarrow d_j) = P(d_j|d_i) \approx \frac{n_{ij}}{n_i},$$

Where nij is the number of sessions with both di and dj.

Ni stands for the number of sessions that contain di. This is similar to the proximity I calculated before for Document Networks but with one exception. Here the denominator is just the number of sessions for page di. With DN, the denominator was the number of documents for ki and kj minus the intersection of the two. This gives an entirely different value for the confidence factor than for the DN proximity measure. Since the denominator of the confidence factor is smaller than for the proximity in the DN, the confidence value is larger than the proximity. And, then the distance being the inverse of the confidence or proximity means the distance will be less for the confidence factor and greater for the proximity. The shortest distance indicates the closer relationships. So, the confidence factor represents closer relationships than the proximity function. And, the difference is not linear comparing confidence to DN proximity. DN proximity could be similarly calculated to see how it represents the DN associations compared to the base proximity computed earlier. Our experiments do this to draw a comparison. [64] So, instead of proximity converted to distance, they use a measure called confidence which is similar to proximity. Then, all the rules are based on the confidence calculations.

Document Classification in a DN using Graph Model

In recent years, classification approaches for documents in a document network have been based on graph models using graph probing techniques such as HTML parse information, and hyperlink and content order information. These techniques extract numerical features from graphs such as node degrees or edge label frequencies rather than comparing the graphs themselves. Better approaches with graph models use graph created solely from the content and use the graphs themselves to determine document similarity rather than a set of extract features. In conceptual graphs based on content, terms and concepts appear as nodes. The edges contain meaning oriented relationships between the concepts expressed in the documents. This is what I refer to as the predications in the Semantic Medline database. So, our case study here utilizes this approach to the graph model in order to base it on the content expressed in each document. Semantic Medline has prepared the database precisely to facilitate this approach.

The classification of graphs using such content based graph model leads to recommendation being provided under the label of graph matching. With this model, there exists a database of graphs and an input (or query) graph with the goal being to find graphs in the database which most closely matches this input graph or query. It is not necessarily expected that the input graph is an exact match to any database graph since the input has not been previously considered. This is referred to as error-tolerant or inexact graph matching since it cannot be

exact. One early approach to this kind of inexact graph matching can utilize the k-Nearest Neighbor method.

K-Nearest Neighbors Method for Graph Matching

For vector space, k-NN is computed from the Euclidean distance from the training input instances in order to determine the classification. The distance equation for vectors is:

Equation 6 - Euclidean Distance [71]

$$dist_{EUCL}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

With graph models, we need a distance measure between graphs to accomplish a similar determination for graph matching. [71] There are a variety of methods for measuring this graph distance using k-NN which depend on the size of the subgraphs. I have already discussed some approaches. Here is another for distance between graphs now. The distance equation for graphs is as follows:

Equation 7 - Distance between Sub-Graphs [71]

$$dist_{MCS}(G_1,G_2) = 1 - \frac{|mcs(G_1,G_2)|}{\max(|G_1|,|G_2|)}$$

Where *G*1 and *G*2 are graphs, *mcs*(*G*1,*G*2) is their maximum common subgraph, max(...) is the standard numerical maximum operation, and |...| denotes the size of the graph which is considered to be the number of nodes and edges. How could I use this information to help rank documents for a given

concept search in a document network like Semantic Medline? Well, I can compute these graph distances as well as the other distances. And, this equation does provide a useful basis for subgraph comparison then in terms of distance.

The graph distance typically outperforms the vector distance in terms of speed according to some studies. It is good to test various approaches to a document network to see how the results may vary with different methods of measuring distance as proposed by research. And, varying the subgraph size also has an impact on the distance and execution time. Graph distance is shown to outperform vector methods. So, this result then indicates improvement in approach once I move away from vector methods and use more the graph approaches to classification and clustering. Therefore, I present graph methods including k-NN for the matching process inherent to Recommender Systems since it does appear that vector methods are not as accurate as graph methods for analysis of distance. Distance helps us to evaluate the strength of document network relationships in a graph database.

Graph Classification and Clustering Algorithms

Graphs are flexible because there are no restrictions on size or labeling to constrain the representation of the graph. Graphs also allow for adoption of size and complexity which can represent the patterns to be modeled. However, there is little mathematical structure in the graph domain as a result of the flexibility. The math operations available for vectors do not exist for graphs and cannot be defined in a standard way across all types of graph data. Therefore, most of the

common methods for mining, learning, and pattern recognition cannot be applied to graphs without modification. They also suffer from their flexibility because whereas with vectors where the distances between object pairs are linear with the number data items, graphs in general are exponential with the number of data items. So, graphs make pairwise comparisons inherently complex and expensive computationally. As a result, there are far fewer algorithmic tools available for graph pattern recognition. [74]

Despite the adverse math and computational conditions for graphs, various procedures for evaluation of graph dissimilarity have been studied in the literature. These procedures are again referred to as graph matching where the overall goal is to find a correspondence between similar substructures between pairs of graphs. And, based on the matching found, a dissimilarity or similarity score can be provided which indicates the proximity of the graph pairs. [74]

Due to these inherent variabilities of the graph patterns being considered and because of the noise associated with graph extraction processes, it is recommended to include in the graph matching framework a tolerance for errors. Graph edit distance offers a way to integrate error-tolerance into graph matching and it has proven to be very useful for graph matching. The key idea with edit distance is to model structural variation by edit operations reflecting modifications in structure and labeling. The main advantage of edit distance is the high degree of flexibility which makes it useful for all types of graph data sets, and it allows for

inclusion of domain specific information about graph similarity by using cost functions specific to the graph type. [74]

Algorithms for computing edit distance between pairs of graphs can be computationally expensive as has been discussed with large number of nodes due to their exponential nature. To make the edit distance less expensive, less optimal methods are often used. One such approach is called Munkres' algorithm which was originally developed for assignment problems where the cost function assigned to elements of two graphs is minimized. This turns the problem of graph edit distance into an assignment problem where nodes and edges can be inserted, deleted, and substituted independently. The Munkres' algorithm returns an optional assignment solution which then is suboptimal or approximation to the graph edit distance. But, the time to solution is less complex and is a cube of the number of nodes in the graphs and is not exponential. [74]

Then, the edit distance can be classified by computing it's dissimilarity to training graphs by using with a k-NN classifier for instance or with kernel machines like support vector machines for classification. This approach has been shown to be very accurate in classification even though a suboptimal approach. This is largely because the distances computed by this approximation method are equal to or larger than the true distances. And, for k-NN classifier, small distances have more influence on classification decisions than large distances. Hence, no real deterioration of the classification accuracy occurs and can be used in place of exact methods which are computationally inefficient. [74]

So, although graph dissimilarity measures can be defined through these kinds of graph matching procedures, it is often not completely sufficient for pattern recognition. A novel approach to overcome limitations of these matching procedures is to use graph embedding into vector spaces. This enables access to the rich algorithms developed for the vectorized approaches not available directly with graph tools. So, new approach with graph embedding procedures can be based on dissimilarity representation and graph matching. The main idea here is to use the distances of an input graph to a number of training graphs called prototype graphs as a vector description of the graph. With this approach, we are using the dissimilarity representation rather than the original graph representation. Then, we obtain a vector space where each axis corresponds to a prototype graph and the coordinate values of an embedded graph are the distances to the elements in the prototype set. So, with this, we can transform any graph to a vector of real numbers. [74]

The method of prototype selection is the main issue with this approach in the embedding framework. The role of the prototypes is crucial because they serve as reference points in the underlying graph domain in order to establish the embedding of graphs. The objective of the prototype selection is to find reference points which lead to meaningful vectors in the embedding framework. There are six basic types of prototype selection as follows:

1. Heuristic prototype selection
2. Prototype reduction

3. Feature selection
4. Dimensionality reduction
5. Lipschitz embedding
6. Ensemble methods

The embedding procedure which is recommended based on advantages discussed before is one which uses graph edit distance because no restrictions for the type of graph exists and due to the degree of robustness against graph distortions. It also supports the use of specific knowledge for the type of graph when defining the cost of the edit operations. Kernel methods can also be used in which the feature space is transformed into higher dimensionality with feature vector spaces. Then, many algorithms are available for classification and clustering to make use of both the flexibility of graphs and the tools available for vectors. [74]

Experimental verification of this embedding framework with graph edit distance substantiates the effectiveness of this approach for many graph types for both classification and for clustering. So, it would be very interesting to use this approach to help us with our Recommender problem in document networks which are graph databases similar to web document systems. The experimental verification did include web documents and it was proven to be very effective for this type as well. The value for recommendation is to find other graphs which fall into the same cluster or classes as the ones being explored by the user. These graphs can also include indirect associations in the graph. So, when requests are

received for recommendations regarding a query which has a graph associated with it, the other graphs in the same cluster or class can be found to display based on the edit distance from the request graph. This would certainly add value to the recommendation being sought by the user and include a variety of novel relationships not previously found with current approaches. But, it is not clear exactly how to use these concepts would benefit a recommender system which needs to respond quickly to requests and which needs to have a given query already somewhat packaged in terms of the best recommendations to make. Pre-clustering for all types of queries is not really feasible with the large database size of Semantic Medline unless it is accomplished when the data is entered into the system. And, this would not be feasible for our research and experimentation here since I do not have the resources to undertake such a huge project.

Similarity Measures for documents in a Document Network

So, given the idea that something like a page rank approach might be best for measuring similarity between subgraphs in a graph database, what are some of the best ways to measure similarity? We have covered graph and node distances previously. There are other ways to measure similarity in the literature as well. Then, once we have a good measure, we could use it for the recommender method similar to page ranking for a search algorithm for web pages and see how that might perform. [72]

This leads to many challenges such as following:

1. Proposing meaningful metrics to capture different graph structural patterns.
2. Designing algorithms which calculate a similarity measure for these metrics.
3. Finding ways to scale these algorithms for large graph size.

Intuitively, we understand that similarity involves graphs having the same or similar nodes and edges in terms of weights to their neighbors. This intuition leads to the possibility of using belief propagation (BP) as a method for measuring graph similarity because of the dependence on neighboring structures. When you include context in the similarity measure, we might infer that graphs for a given system would have more similarity depending on contextual factors such as time frame or in the case of medical document systems like Semantic Medline depending on the researcher's profile. So, context can be built into these kinds of similarity measures as well. [72]

From the perspective of subgraph matching, consider a series of graphs over the same set of nodes but with different edges. This is what we have in Semantic Medline where there are references to documents with the same concepts or nodes but their connection to the objects (predications or edges) are different. We would like to identify these subgraphs and assess their similarity between one and the other for purpose of ranking them to the original search predication. Then, with this similarity measure, it can feed directly to a ranking algorithm to recommend the most similar neighboring predications or subgraphs

to the original. This can possibly lead directly to our dissertation objective to find the best approaches for finding recommendations in a document network like Semantic Medline. [72]

Some of the basic approaches to measure graph similarity are as follows:

1. **Edit distance** – The graph edit distance is a generalization where the goal is to transform one graph to the other by doing operations of nodes and edges where each operation has a cost. The sequence of operations is minimized in order to match the two subgraphs. The main problem is the algorithm is exponential and does not work well with large graphs.
2. **Feature extraction** - The idea here is that similar graphs probably share certain properties such as distribution degree, diameter, and eigenvalues. After extracting these features, a similarity measure is applied to the aggregated statistics. These methods scale very well with large graphs. However, depending on the statistics chosen, it is possible the results are not very intuitive such as getting a high similarity measure between different graphs with very different node set sizes. So, it is not great for all kinds of graphs.
3. **Iterative methods** – The philosophy here is that two graphs are similar if their neighbors are similar. For each iteration, the nodes exchange similarity scores. The iterative process ends when convergence is achieved. SimRank is one such algorithm which

measures self-similarity of a graph and is based on the idea that similar nodes have similar neighbors. It computes all pairs of similarity scores by propagating similarity scores in an adjacency matrix of the graph and ends when it converges. There are many other iterative methods which have merit.

So, there are actually quite a number of research papers on this topic and they all seem to end with the conclusion that the use of normalized version of Euclidean distance is very intuitive measure of similarity. The distance measures we have discussed previously are in fact Euclidean distance measures as well. And, PCA is the most desirable and the fastest approach for performing subgraph matching along with mining of local graphs for clusters using Markov Clustering algorithms produces highly relevant results for large graphs. [72]

Here we are with Markov algorithms again which applies to the page-ranking algorithms as well. It looks like we need to have Markov included in our experimental options. But, it appears the page-ranking approach has more value in our case study here than clustering because we need to build ranked lists of documents for recommendation and that depends on a user query submitted ad hoc without sufficient time to build clusters for the query. And, building all of them in advance with a huge graph database as is in this case study is not realistic. The document ranking needs to be able to run quickly and improve as they are run multiple times. And, this is precisely how the page ranking algorithms run. But, this study of similarity is important as additional background for

determination of how to associate documents as part of a page-ranking algorithmic process. [72]

Similarity or distance is appropriate measures to use in a document ranking algorithm for our case study. I can try different methods to see which ones have the best results using evaluation approaches appropriate for Recommender Systems discussed in this Appendix. I now discuss some of these approaches based on the previous research presented so far.

Context in Recommender Systems

Various aspects of the recommendation use case will be considered as relevant. One of those aspects is the context of the desired recommendation. Context is the user's specific situation and environment at the specific time of using a web based application or other automated information system service. Context will change over time so there is clearly a specific time aspect to context. And, context will change based on changes in the user's situation and environment over time. So, it is not a static phenomenon. But, context can be a predictable phenomenon for a user based on past behavior learned for based on specific content at specific times and within specific situations and environments. So, the challenge of the inclusion of context as a series of attributes for the user to improve the applications ability to interact with the user is based on the accumulation of a context model for the individual in relation to these learned situational and environmental behavioral factors. Then, to invoke the model at a given time for a given situation, it is necessary to make intelligent judgments as

to the user's specific situation when using web-based applications in order to invoke the correct learned context for the user at the time of interaction with the application.

What is context in Recommender Systems and a provide summary of what it means in relation to information systems? Then, I will build on this understanding to present approaches to include the use of context to improve the information provided through automated system interactions. Finally, I will answer some questions posed as to how this body of knowledge may be utilized and discuss areas where gaps in knowledge seem to exist for additional research to benefit the overall understanding.

Define the notion of "context" in regards to this research: Context is the user's specific situation and environment at the specific time of using a web based application or other automated information system service. Context will change over time so there is clearly a specific time aspect to context. And, context will change based on changes in the user's situation and environment over time. It is not a static phenomenon. But, context can be a predictable phenomenon for a user based on past behavior learned for based on specific content at specific times and within specific situations and environments. The challenge of the inclusion of context as a series of attributes for the user to improve the applications ability to interact with the user is based on the accumulation of a context model for the individual in relation to these learned situational and environmental behavioral factors. Then, to invoke the model at a

given time for a given situation, it is necessary to make intelligent judgments as to the user's specific situation when using web-based applications in order to invoke the correct learned context for the user at the time of interaction with the application.

For example, if a user utilizes web-based applications during working hours, their context may be more work oriented as it might relate to interactivity with the application. Instead of being interested to purchase a book for the office more likely during working hours, the user may have a more personal business context after working hours to purchase books for pleasure. An Amazon search for books may be more work oriented during working hours and more personal interest oriented during non-working hours. But, certainly there will be considerable overlap for something like this as people will do personal searches during working hours at times. It will be more probabilistic and less deterministic when factoring context into the equation for application interactivity unless the user's current context situation at the time of interaction can be provided. For example, the application could ask questions on initiation of the interaction or request user profiles to be invoked to establish their specific context. Certainly, when doing searches, the user would wish to provide context information to enhance the search. But, if not an active interaction and the context is more driving advertising for example, then the more probabilistic approach will need to take over as the user would not likely identify ads they wish to see. The application would need to figure determine context on its own.

There are a number of characteristics associated with context. Context can be considered to be a set of constraints to influence a given system. As in the example above, certain products are likely to be of interest depending on the time of day. The constraint is time. Context is a collection of attributes to provide preference, perspective, and approximation to intent of the user. Context influences behavior and is formed by perception, memory, and reasoning all of which can depend on many variables. Context can be considered to be a state of mind with few hard boundaries consisting of many elements all of which are dynamic in nature. Context consists of rules and resources which regulate interactions of the user. Therefore, to model context accurately will depend on many attributes and the model will become quite complex and will clearly be probabilistic and not deterministic. **[18]**

From a more general perspective, context is the information that can be used to characterize the situation of an entity. An entity can be a user as discussed so far. It can be an entity with which an individual user needs to interact as well. It is quite possible that the context of multiple entities would need to be considered in use cases where more than one entity context is important to the outcome whether it be a search or a recommendation. In the Zimmerman article on “Operational Definition of Context”, the elements of context are individuality, activity, location, time, and relations.

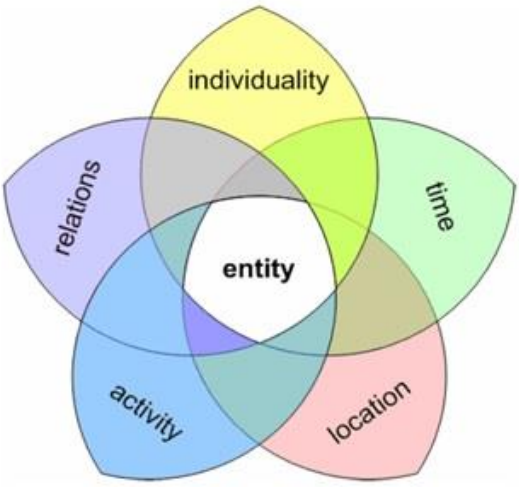


Figure 31 - Components for User Context [18]

These bigger categories of elements divide into personal elements and environmental elements with 12 elements in dimension space, 4 in spatial-temporal, and 8 in human intent. Within the Individuality Context, there are several elements to consider. [18]

From the research, the principal techniques to improve the recommendation involved pre-processing the data to eliminate outliers, elimination of data noise, and reduction of unwanted global dataset effects. Also, prioritizing dimension and reducing the number of dimensions is very valuable to focus on the pertinent data characteristics which are best suited to make recommendations in a given case. In our chosen Document Network of Semantic Medline, a great deal of pre-processing is accomplished with the database in order to pre-process the data for best recommendation results.

a) Information discovery (search);

An example of the use of context to improve information discovery is found in the area of e-learning. First, in the case of searching for e-learning products, in order to provide the needed information which might be based on context requires intelligent methods for representing and matching learning needs with learning contexts. One framework proposed for this application includes a semantic e-learning domain of course concepts in lecture ontology. Also, this needs to be combined with learner profiles using navigation logs that have recorded lectures which have been previously accessed. Documents are clustered to discover sub-concepts which may not be included in the available course taxonomy. Then, with this structure in place, the learner's search results are ranked based on matching concepts of the user profile with the learning content to provide ranked citations. The learner is provided with semantic recommendations in the search process in the form of terms from the closest matching clusters in their specific profile. [18]

The framework to achieve this method of search for the e-learning case has 3 layers: (1) semantic representation (knowledge representation), (2) algorithms (core software), and (3) personalization interface. The following figure provides a graphical representation of how this framework would be structured:

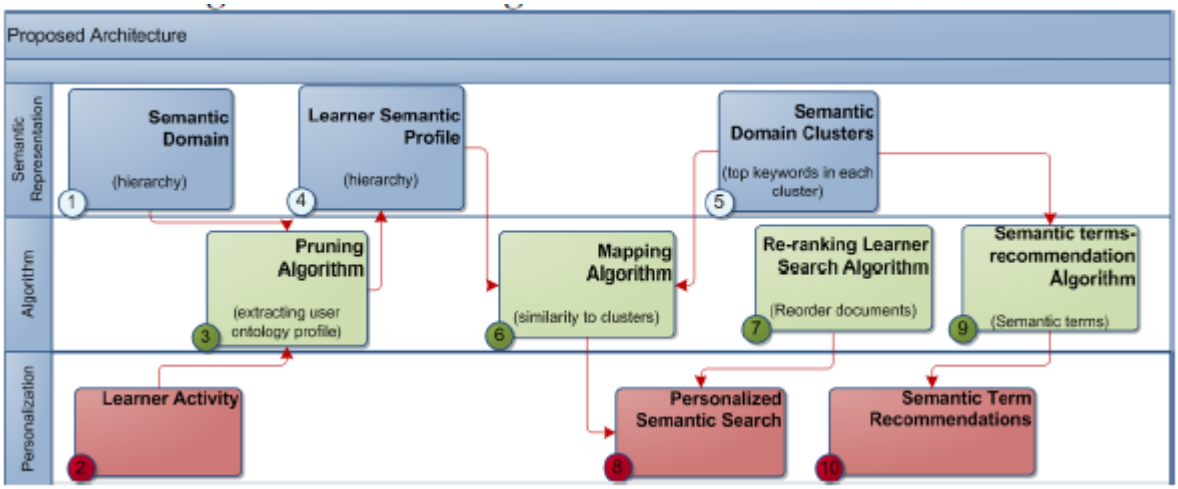


Figure 32 - User Context Architecture Framework [14]

The algorithm utilized is cluster-based and depends of class document clusters to identify the correct relationship to the best semantic representation. It also includes pruning, mapping, ranking, and semantic terms recommendations. Similarly, the learner's semantic profile is built from the ground up based on document historical visits and pruning algorithms to get the tree structure constructed well. Then, there is a document cluster to learners profile mapping to connect the semantic domain clusters to the personalization. Experiments with this approach show that the learner's context can be effectively used to improve search precision in ranking based on the learner's past learned activities. **[14]**

Another proposed framework for an ontology-based approach to semantic context-aware e-learning search uses knowledge about the learner, about the content, and about the domain. This results in a personalized, complete, and augmented learning program for adaptive content recommendation. This work is

different from other approaches because it based on knowledge-based semantic approaches. It makes use of three ontologies:

1. Learner Ontology
2. Learning Content Ontology
3. Domain Ontology

These ontologies may appear as follows:

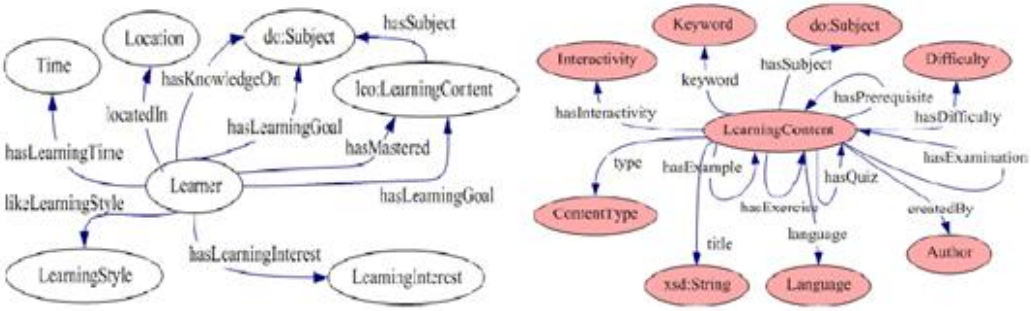


Figure 33 - User Context Ontology Models [20]

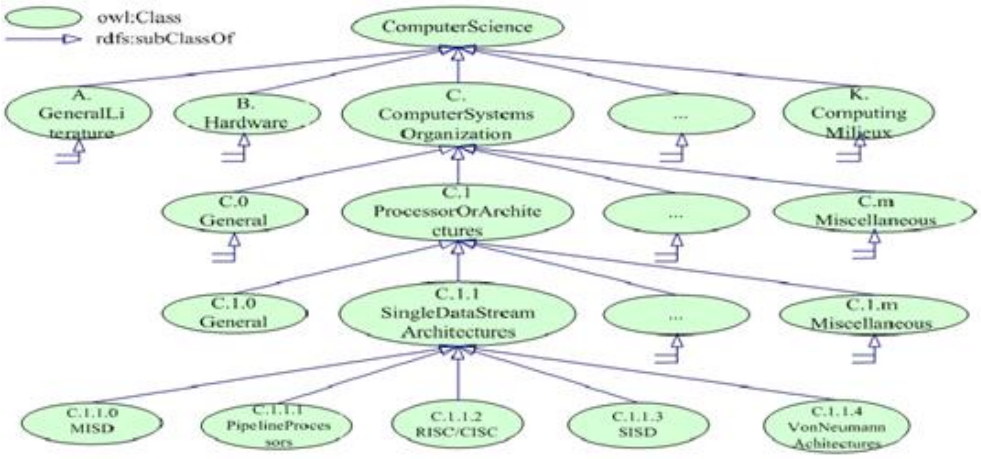


Figure 34 - Computer Science Domain Ontology [20]

The search approach has 4 steps:

1. Semantic relevance calculation
2. Recommendation refining
3. Learning path generation
4. Recommendation augmentation

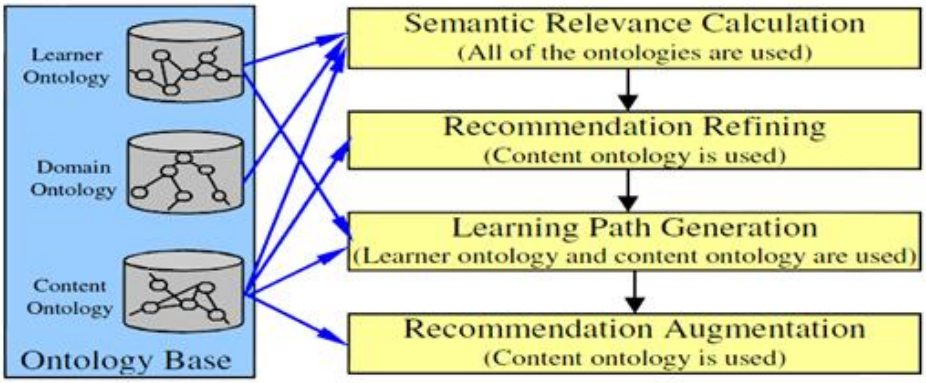


Figure 35 - Content Recommender Procedure [20]

This approach provides a good example of how to actually implement an ontology-based e-learning search recommender system. Experiments with this framework have shown it is light-weight and feasible to be deployed with good response time. For future work, additional learner contexts need to be included for available learning time, location, learning style, and learning interests to make the system more comprehensive and intelligent. It is also important to consider the shared-knowledge among group members so as to recommend content to a group of learners. But, it does illustrate how semantic frameworks can be useful to make better recommendations in search. **[20]**

b) Decision making (recommendations);

Current generation of Recommender Systems can be classified into three main categories:

1. Content-based
2. Collaborative

3. Hybrid

These systems have current limitations and extensions which would have promise. One of the main extensions is the inclusion of contextual information in the recommender process and other areas. These extensions are necessary in order to make decision making more effective.

The utility of a recommendation is valued with a ranking or by a profit value. The user space can be defined with a profile of user attributes such as age, gender, income, marital status, etc. Utility is defined for a subspace of the user included with the object. With a given rating, the model attempts to project to other area the preferences there. But, it is not always transferable. Therein lays the problem with current Recommender Systems. Their ability to project to other areas is very weak. Extrapolations are normally done with very poor models and estimates are made which are far from perfect. New methods of extrapolation need to be found to improve the predictive processes. **[22]**

Content based recommender systems utilize user profiles which try to model the user tastes and preferences from weighting keywords. There are a number of measures for this but none are very accurate. So, if a user reads a lot of a particular type of article, then the ranking for those kinds of keywords have higher value in the utility function. Also, Bayesian classifiers are used in content based systems which have better accuracy. Still other approaches are called adaptive filtering, and threshold setting. Content—based techniques are limited to the features explicitly stated with these objects. Some systems have problems

with automatic feature extraction. And, if two different contents are identified with the same features, they then become indistinguishable. **[22]**

When the recommendations are limited to the scores of previously rated items, then that constrains other kinds of recommendations. But, this is how many recommender systems function today. With collaborative methods, the system looks for other users who have rated similar content and they use their ratings for new content of interest to the original user. These are generally known as memory based collaborative techniques. One approach for this type of recommendation is to calculate all the values in advance of the next recommendation. Then, utilize that database of vales for new recommendations. **[22]**

In contrast to content and collaborative approaches, model –based approaches attempt to learn a model which is then used to make rating predictions. There are two types of such models; cluster models and Bayesian models. In cluster methods, like-minded users are clustered into classes. The user ratings are assumed to be independent. There are a number of these modeling approaches including latent semantic analysis. The main difference is these model-based approaches are not using rules but more statistical and machine learning approaches to create the model. But, this approach resolves the problem with the other approaches which depend on prior ratings stored for historical purposes. **[22]**

t is clear the model-based approach is best combined with context elements to create a model for a given user which can be utilized for new items and not based solely on historical memory of the ratings of previous items. A semantic model would be an excellent way to provide a much more highly accurate model. **[22]**

Another framework suggests the use of semantic context in recommendations to leverage arbitrary background information relevant to the process. A modeling framework is presented for a wide class of semantic recommendation tasks. The framework includes decomposing and optimizing tensors by Bayesian personalized Ranking (BPR) criteria. Training data is used to define the multi-sets for the individual relationships between context and the entity. Then, the process would interpret a list of recommended entities based on the ranking from the training data. Next, the resulting probabilities are optimized and the tensors are decomposed. RDF is recommended as an input to the overall process to improve the overall accuracy. This approach has real value but it does have accuracy issues and the use of RDF would improve the process. **[9]**

With still another approach, a series of ontologies are proposed including a user model, GUMO (General User Model Ontology), multimedia content and context ontologies, and a rule-based matchmaking approach for recommending systems. Category-based preferences as well as the expression of any interest concerning a concept formalized in ontology are allowed. Furthermore, recommendations can be provided adequately in different situations as the user

ontology allows expressing context-dependent interests. The approach has been successfully applied in a prototype of video recommender for mobile device. The representation of these interacting ontologies are shown below: **[15]**

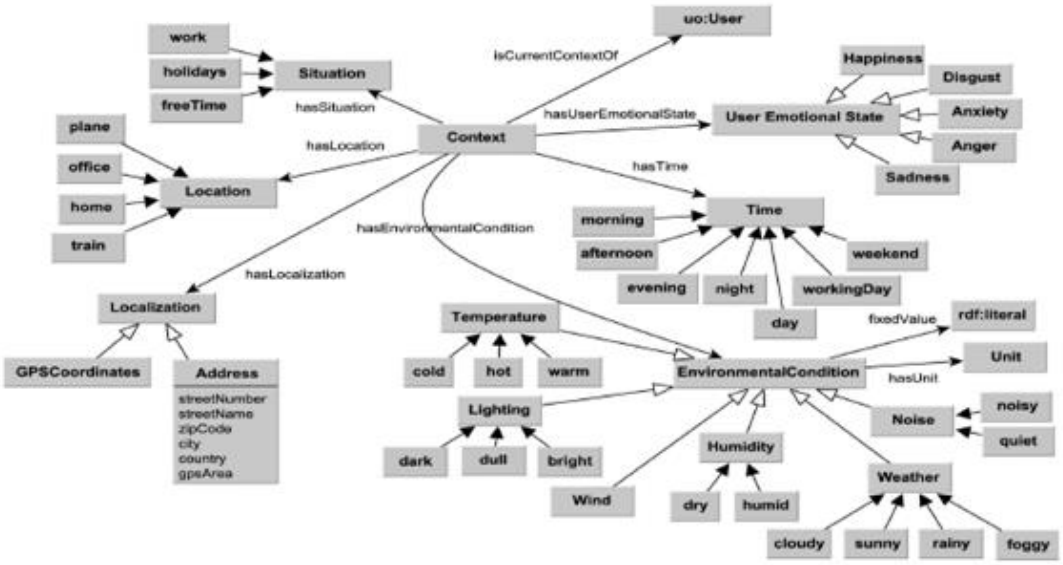


Figure 36 - Integrated Context Ontology Diagram [22]

These ontologies present an interesting approach which has good merit and potential accuracy for recommendations. The proposed architecture to implement such a system of ontologies is as follows:

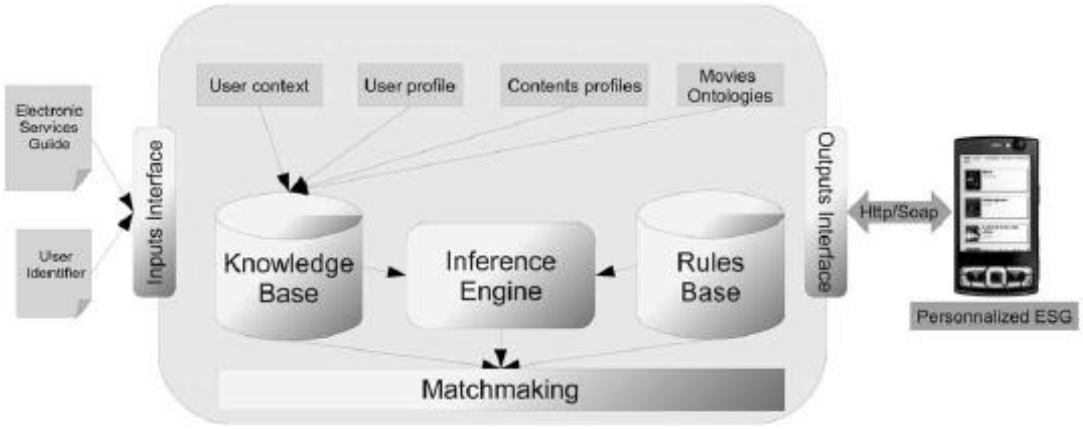


Figure 37 - Recommender System Architecture with User Context [22]

This approach has been successfully applied to recommendations for TV content. It is a more generalized model which can apply to other content as well. The results obtained with this approach have been good and areas where gaps exist have been identified for future work.

Another research paper presents a personal interpretation of the evolution of artificial intelligence (AI) systems during these last 25 years. This evolution is presented along five generations of AI systems, namely expert systems, joint cognitive systems, intelligent systems, intelligent assistant systems, and the coming generation of context-based intelligent assistant systems. The research discussion relies on different real-world applications in different domains, especially for the French national power company, the subway companies in

Paris and in Rio de Janeiro, in medicine, a platform for e-maintenance, road safety, and open sources. The main claim of the research is to underline that the next generation of AI systems (context-based intelligent assistant systems) requires a radically different consideration on context and its relations with the users, the task at hand, the situation, and the environment in which the task is accomplish by the user; the observation of users through their behaviours and not a profile library; a robust conceptual framework for modelling and managing context; and a computational tool for representing in a uniform way pieces of knowledge, of reasoning, and of contexts. **[22]**

c) Social Networks (Facebook, Twitter, Google+, etc.);

Systems for annotation within social networks provide an organization of information according to user-defined keywords. This framework does provide a method to discover, organize knowledge, and to connect to other users with similar interests. However, these systems are not scalable to large social networks so system performance is a major limitation to fully implement such approaches. Because of this fact, other techniques have been considered and tried. So far, those techniques have focused on tag recommendation. The area of resource recommendations in social networks has not as yet been fully explored. One approach is to provide a linear-weighted hybrid framework for resource recommendation in social networking. It has been shown using real world datasets that this integrated approach is essential to the recommendation task and that it provides the best adaptability given the different capabilities of

different social systems. They have found that this approach is more effective than more complex mathematical techniques in practice while providing greater flexibility and extensibility. **[4]**

Another article proposes a framework for the inclusion of social networks for applications to integrate attitudes into the user domain. This approach involves the use of social data clustering methodologies. These methodologies typically involve sentiment analysis or opinion mining. This refers to the study of opinion sentiment using computational techniques to study the emotion of the resource. The framework presented for this Dissertation is as follows: **[3]**

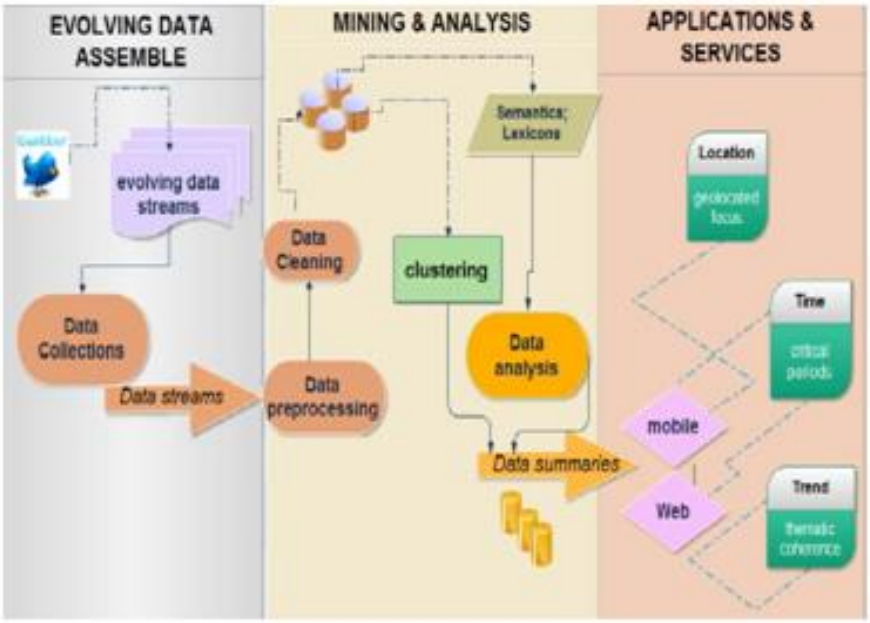


Figure 38 - Framework for Minina and Analytics [3]

Another framework in the research examines strong versus weak semantic techniques – Strong = semantic web techniques (tools for building and querying ontologies; Weak = annotation techniques gets less attention. Tagging systems are based on personal points of view and help with collaboration. Tagging produces overlapping structures. Ontologies are more about knowledge hierarchies. Tagging is expensive and difficult to align with ontological approaches. This framework is a result of the discussion of these semantic techniques and offers another framework for a more generalized model to assist social networking.

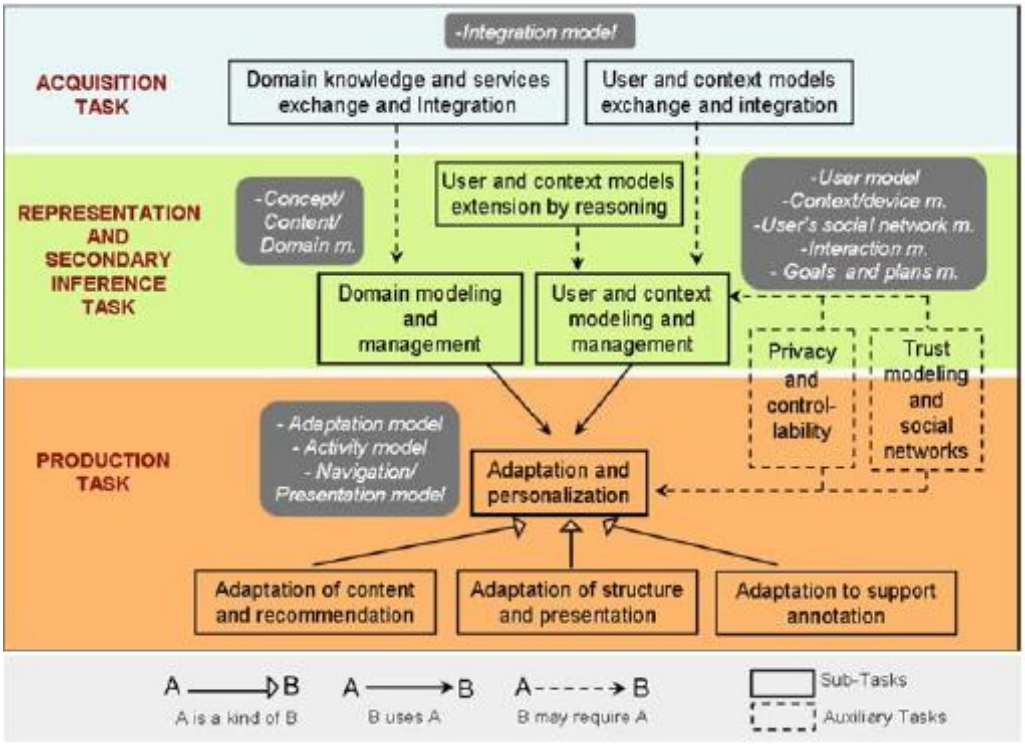


Figure 39 - Functional Model of Recommender Tasks [12]

Note the user and context models are at the core of such a system framework. The results of this approach indicate that strong semantic techniques are better suited to tasks associated with domain and user knowledge exchange, integration, and reasoning. Also, it has been clearly concluded that social tagging can greatly improve user models. Also, it has been clearly shown that mixed approaches which combine semantic and tagging techniques can result in the advantages of both for a given task. **[12]**

Another research project presents what they call the SPETA system. SPETA uses knowledge of the user's current location, preferences, and past historical locations as the basis for recommendations for tourists during a tour. The objective of the study was to build a better user experience for tours. The strength of the approach was again based in combining a context-aware system with a GIS system using social networks and semantic techniques. The results of this approach were quite important to further the notion that semantic inclusion in recommender systems does have overall value to the process. So, here is another citation which goes to using semantic ontologies to support decision making processes. **[13]**

Association retrieval for learners in online social networks has also proven to be an excellent approach to support recommender systems. With this approach, learners in the social network share their resource ratings with their friends in the social network. Similarity between friends is derived from the resource ratings over history. Association retrieval techniques are employed to infer recommendation scores for resources with a given learner. Experimental results show this approach outperforms collaborative filtering in benchmark tests. So, with the huge growth of learning materials and resources in social networks, finding suitable materials or resources based on learner's preferences including their learning styles and knowledge levels is extremely challenging. E-learning recommender system (eLRS) provides a great approach to solving the overload problem by providing valuable resources to learners. **[2]**

Another proposed approach combines semantic web technologies with linguistic values as a more implicit way of doing context search recommendations. A Friend-of-a-Friend (FOAF) model is link with various other approaches such as RDF (RDF-Personal Info Markup), FOAF associated with weighted interests, geonames database inclusion, RDF geographical coordinates to relate to a region, Temporal Thing for date or duration, abilities or disabilities, and user activities and roles model. Combine FOAF with user linguistic values using an aggregation score. This user model appears graphically as follows: **[8]**

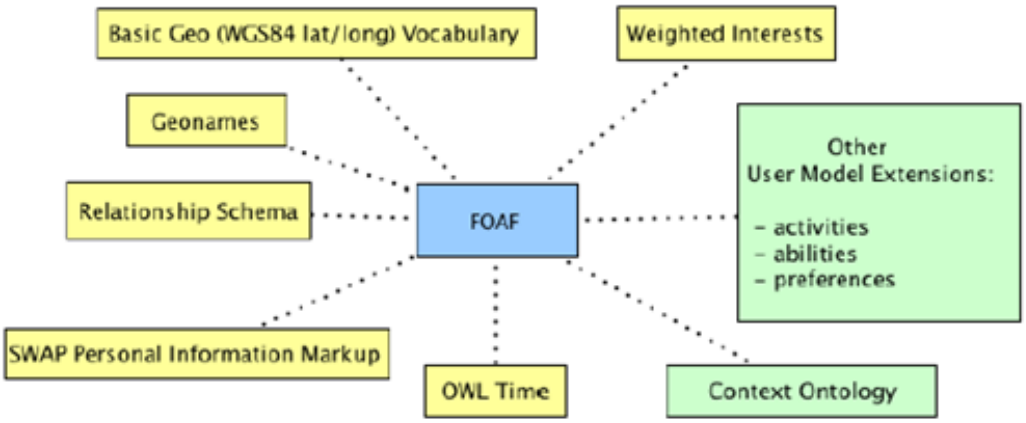


Figure 40 - eFoaf Semantic User Model [8]

This approach apparently needs a lot of validation and further work to really fine tune the concept and make it into a real recommender system. But, the idea is good and it further points out the importance of the RDF model for use in generalized user models for recommender systems.

d) Advertising (product/ad placement).

The travel system discussed in the article utilizes a semantic RDF database of user interests. It is an example of a semantic recommender system. There are several semantic filtering processes utilized as follows:

1. Content based filtering
2. Collaborative filtering
3. Knowledge-based filtering
4. Hybrid filtering techniques

The third element here is a semantics-based service. A prototype was developed to examine the results of such an approach. It uses OWL-based user semantics to discover travel services, and RDF-based user semantics to match user interests with services. The system architecture looks like the following:

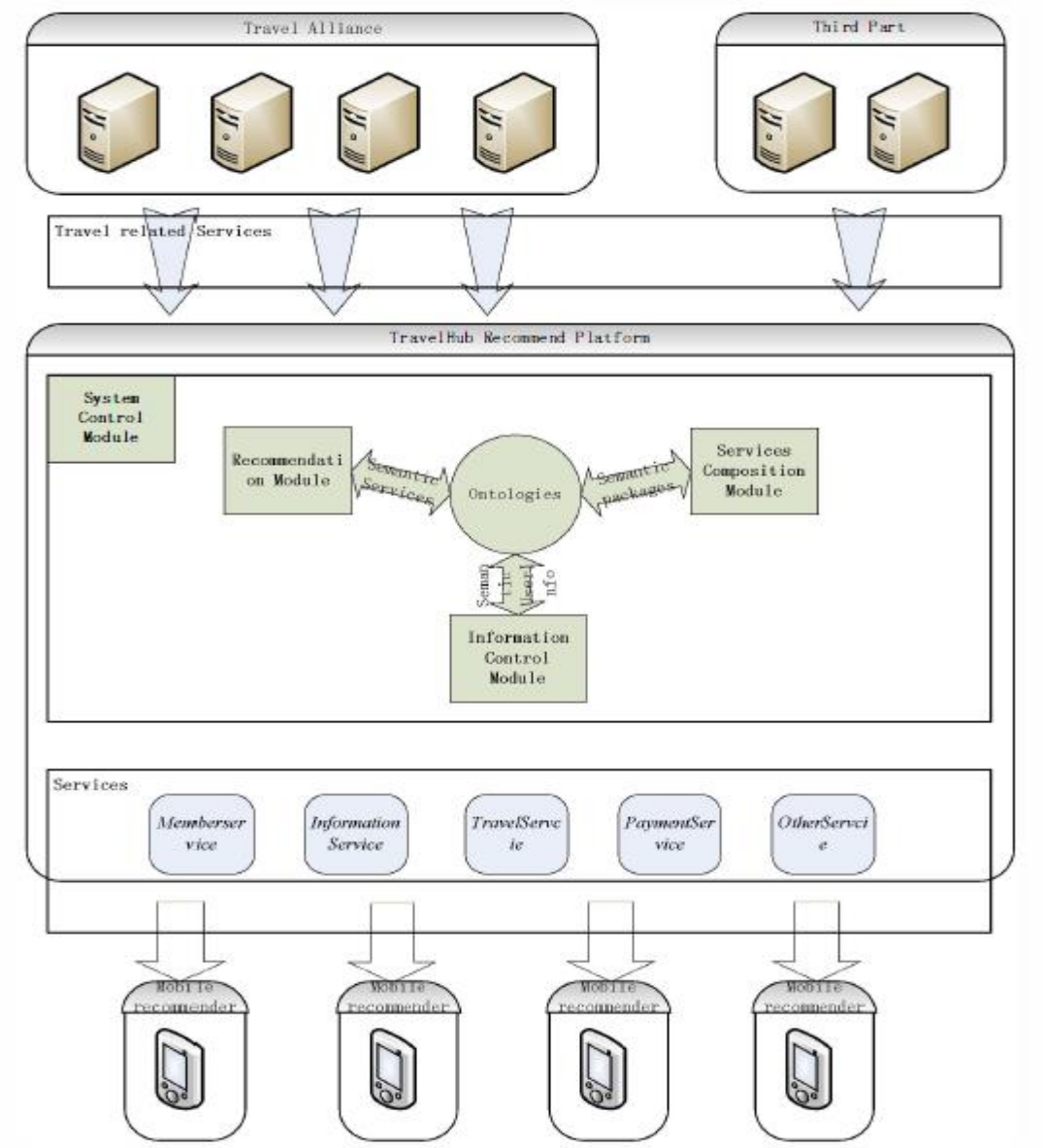


Figure 41 - Recommender System Platform Architecture [5]

The prototype has been very successful to see how such an approach could work. But, the actual results are not as impressive as expected so the approach needs much more research and development to improve performance.

But, here is another instance of semantic techniques being integrated to the design which again has positive outcomes when tested. **[5]**

Still another approach combines enterprise-product semantic model with a user context ontology model and have a way to map between the two. The paper proposes an architecture for the approach CACRIS. It discusses all aspects of this architectural approach which appears graphically as follows: **[6]**

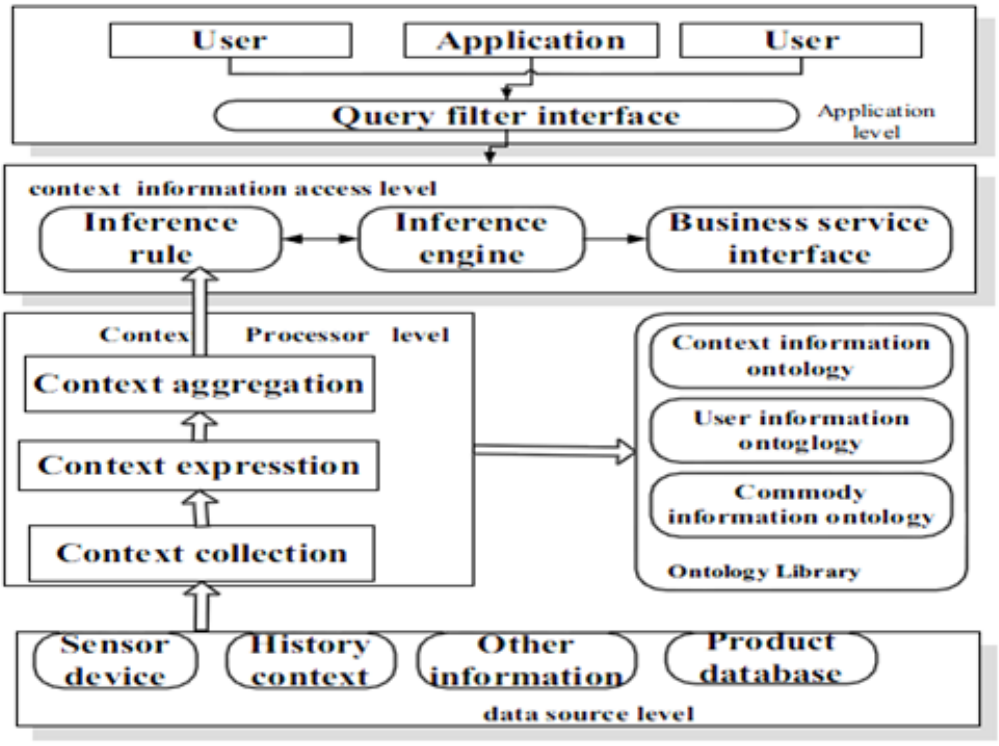


Figure 42 - Context Aware Recommender Platform [35]

This framework shows how to use semantic ontology to match commodity information with user information. A semantic ontology-based context-aware algorithm matches the relationships. This approach has shown to be very effective with high efficiency and accuracy in real world applications. This approach is also very scalable for large-scale, ontology-enabled recommender systems for e-commerce.

How can we acquire information about a person's 'context'?

With a semantic framework to model users, context-related information is gathered about the users. The perceived context data is lifted semantically into context ontology. The context data is time stamped and included in the user history ontology. This enables the application to construct a timeline of events and actions as part of the user's history. Then, these historical records can be analyzed to fine clustered action-patterns which can enable automatic rule definitions. **[35]**

Context information can be gathered explicitly, implicitly, and through inference.

• Explicitly – by asking direct questions or by having users fill in profiles of their interest with forms, explicit context information involves the user's knowledge of the information collection.

• Implicitly – using data in the public domain, contextual information can be gathered from past behavior which searching the internet, past purchases or other historical information of actual behavior which is time stamped and saved

to create a historical record from which context can be implicitly established. The user does not provide the information as it is gathered from other sources based on past behavior of the user.

• Inferring – context is inferred using statistical methods and data mining. Generally, a predictive model is built and it is trained with appropriate training data. The quality of the inference will depend on the quality of the classifier used and its effectiveness will vary across applications. **[10]**

Is it possible to infer context? If so how, if not, explain why not?

It is very possible to make inference based on learned user behavior. I have presented several frameworks in the discussion so far where inference is integral to the design. And, in all those cases, inference did improve the performance of the approach. Bayesian inference is the principal learning model in these cases which is based on probability. However, the use of these methods is still very much in its infancy at this time. The capability is there but it is hampered by the lack of a comprehensive user model which might fit many inference situations and which could be standardized across platforms for a variety of purposes. The current state appears to very much be custom to the specific application. [**32]**

How might we improve on capturing user context, and using it to make better recommendations?

One such system studied in a listed reference is the AMAYA recommender system built to provide recommendation for all situations which

could arise in a user's life, and which can support any number of services a user might want to use. To achieve this goal for all situations, all personalization data for specific situations will have stored mapping. An ontology-based content categorization scheme is provided to provide a service-independent interface.

AMAYA consists of four components which include as follows:

1 – Data Adapter – enables retrieval and processing of distributed personalization data by providing an interface which abstracts the content

2 – Profile Manager – groups the personalization data in terms of profiles which provide a mapping to specific situations

3 – Profile Broker – supports queries of the personalization data for specific situations

4 – Recommender – supports a learning contextual user model for a service-independent interface for the learning algorithms and prediction models.

The learning algorithms will depend on the situations encountered in the learning process and is linked to the current situation as they learn. The contextual retrieval of the model for specific situations enables the recommender. The advantage of this approach is that well-known algorithms can be used directly with little or no modifications. The drawback is all situations have to be explicitly defined by the Profile qualifiers. This makes it much less generalizable. The results are good with news articles is about as far as the research went. So, it was not discussed how would work with other content. But, another example of a system studied which can be improved with user context. **[21]**

Many recommender systems in use today utilize collaborative filtering techniques. Combine this approach with predictive context models of user preferences, interests or behavior found within social networking data, and the accuracy increases substantially. The algorithms used rely on statistical modeling techniques that introduce latent class variables in a mixture model to discover user groupings and interest profiles. The main advantage of this approach was found through experimentation to be in accuracy, excellent time prediction, and good user model representation. So, this is another example of the use of context improving the recommendation. And, actually, all through this paper we have presented many examples of how context has been proven through experimentation to improve the recommendation process. It is just a matter of how it is done and the objective of the recommendation. There are so many examples that the case for context inclusion has been substantiated over and over again. **[26]**

Test Results by Medical Concept

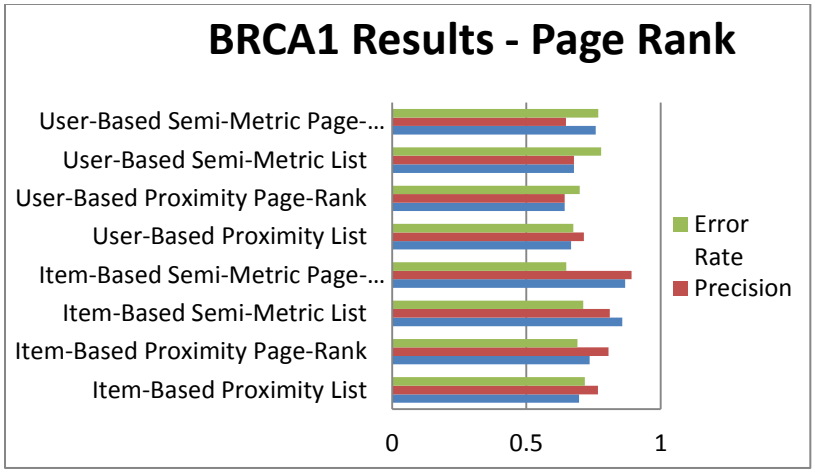
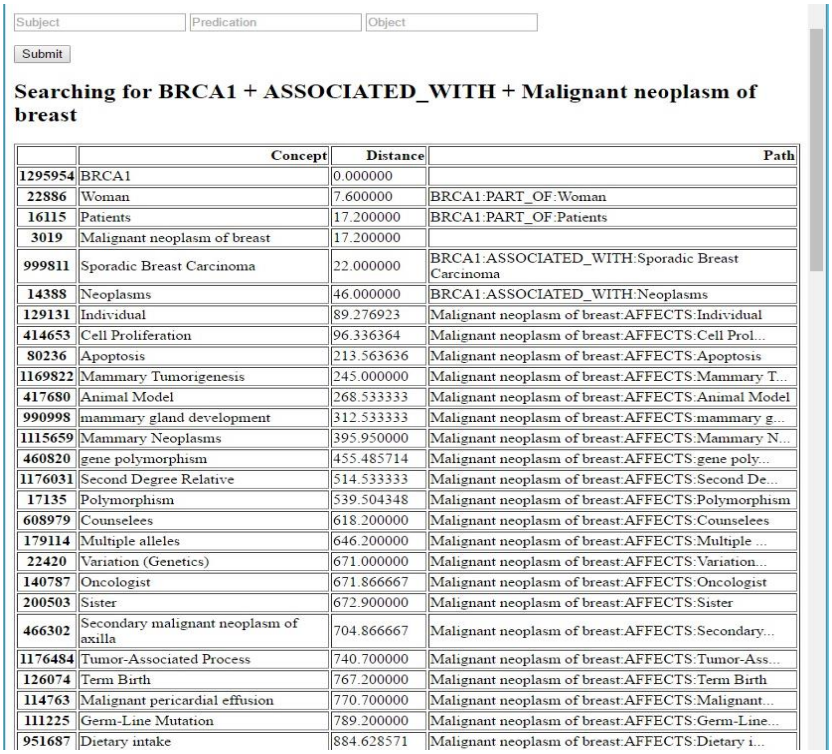


Figure 43 - BRCA1 Page Rank Results Bar Chart



Subject | Predication | Object

Submit

Searching for BRCA1 + ASSOCIATED_WITH + Malignant neoplasm of breast

	Concept	Distance	Path
1295954	BRCA1	0.000000	
22886	Woman	7.600000	BRCA1:PART_OF:Woman
16115	Patients	17.200000	BRCA1:PART_OF:Patients
3019	Malignant neoplasm of breast	17.200000	
999811	Sporadic Breast Carcinoma	22.000000	BRCA1:ASSOCIATED_WITH:Sporadic Breast Carcinoma
14388	Neoplasms	46.000000	BRCA1:ASSOCIATED_WITH:Neoplasms
129131	Individual	89.276923	Malignant neoplasm of breast:AFFECTS:Individual
414653	Cell Proliferation	96.336364	Malignant neoplasm of breast:AFFECTS:Cell Prol...
80236	Apoptosis	213.563636	Malignant neoplasm of breast:AFFECTS:Apoptosis
1169822	Mammary Tumorigenesis	245.000000	Malignant neoplasm of breast:AFFECTS:Mammary T...
417680	Animal Model	268.533333	Malignant neoplasm of breast:AFFECTS:Animal Model
990998	mammary gland development	312.533333	Malignant neoplasm of breast:AFFECTS:mammary g...
1115659	Mammary Neoplasms	395.950000	Malignant neoplasm of breast:AFFECTS:Mammary N...
460820	gene polymorphism	455.485714	Malignant neoplasm of breast:AFFECTS:gene poly...
1176031	Second Degree Relative	514.533333	Malignant neoplasm of breast:AFFECTS:Second De...
17135	Polymorphism	539.504348	Malignant neoplasm of breast:AFFECTS:Polymorphism
608979	Counselees	618.200000	Malignant neoplasm of breast:AFFECTS:Counselees
179114	Multiple alleles	646.200000	Malignant neoplasm of breast:AFFECTS:Multiple ...
22420	Variation (Genetics)	671.000000	Malignant neoplasm of breast:AFFECTS:Variation...
140787	Oncologist	671.866667	Malignant neoplasm of breast:AFFECTS:Oncologist
200503	Sister	672.900000	Malignant neoplasm of breast:AFFECTS:Sister
466302	Secondary malignant neoplasm of axilla	704.866667	Malignant neoplasm of breast:AFFECTS:Secondary...
1176484	Tumor-Associated Process	740.700000	Malignant neoplasm of breast:AFFECTS:Tumor-Ass...
126074	Term Birth	767.200000	Malignant neoplasm of breast:AFFECTS:Term Birth
114763	Malignant pericardial effusion	770.700000	Malignant neoplasm of breast:AFFECTS:Malignant...
111225	Germ-Line Mutation	789.200000	Malignant neoplasm of breast:AFFECTS:Germ-Line...
951687	Dietary intake	884.628571	Malignant neoplasm of breast:AFFECTS:Dietary i...

Figure 44 - BRCA1 Semi-Metric Search List

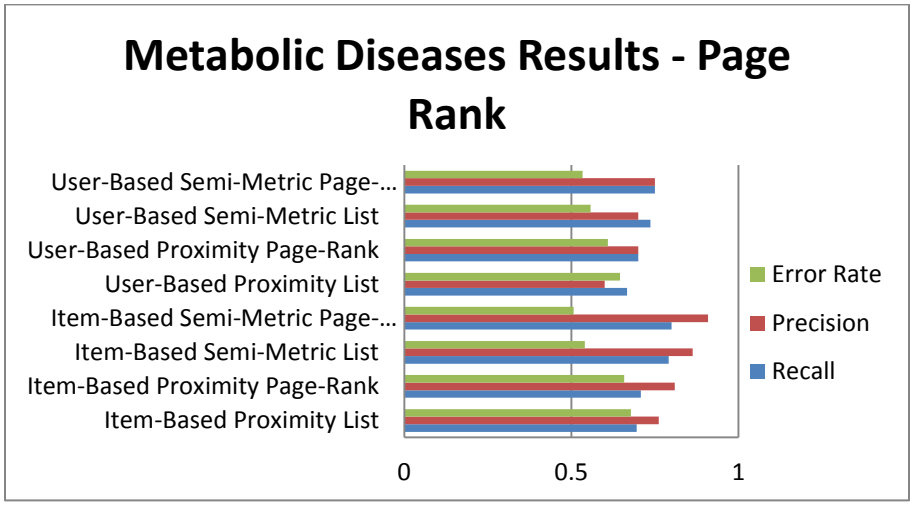
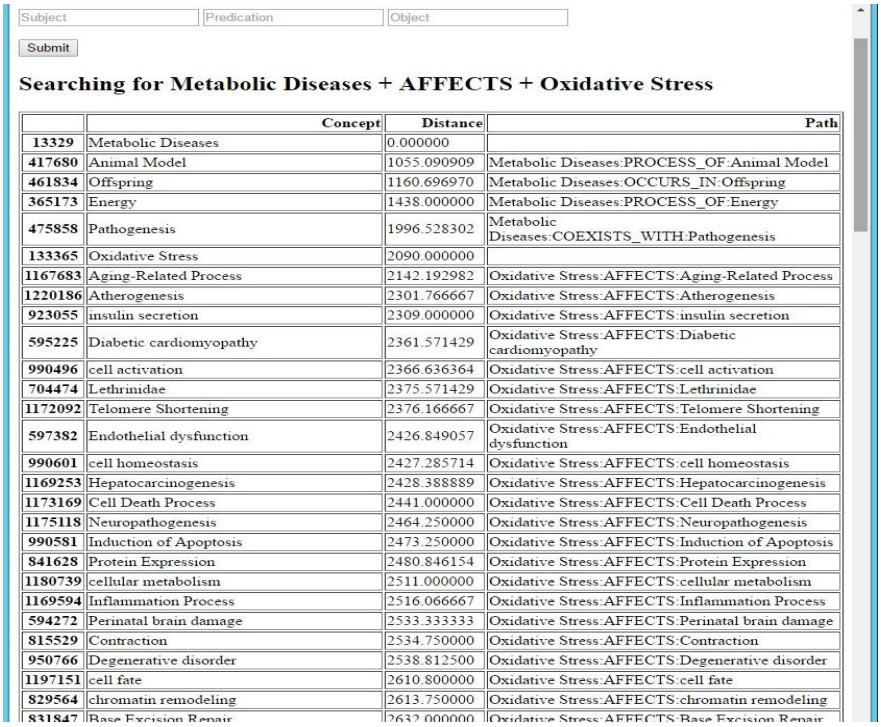


Figure 45 - Metabolic Diseases Page Rank Bar Chart



Subject | Predication | Object

Submit

Searching for Metabolic Diseases + AFFECTS + Oxidative Stress

	Concept	Distance	Path
13329	Metabolic Diseases	0.000000	
417680	Animal Model	1055.090909	Metabolic Diseases:PROCESS_OF:Animal Model
461834	Offspring	1160.696970	Metabolic Diseases:OCCURS_IN:Offspring
365173	Energy	1438.000000	Metabolic Diseases:PROCESS_OF:Energy
475858	Pathogenesis	1996.528302	Metabolic Diseases:COEXISTS_WITH:Pathogenesis
133365	Oxidative Stress	2090.000000	
1167683	Aging-Related Process	2142.192982	Oxidative Stress:AFFECTS:Aging-Related Process
1220186	Atherogenesis	2301.766667	Oxidative Stress:AFFECTS:Atherogenesis
923055	insulin secretion	2309.000000	Oxidative Stress:AFFECTS:insulin secretion
595225	Diabetic cardiomyopathy	2361.571429	Oxidative Stress:AFFECTS:Diabetic cardiomyopathy
990496	cell activation	2366.636364	Oxidative Stress:AFFECTS:cell activation
704474	Lethrinidae	2375.571429	Oxidative Stress:AFFECTS:Lethrinidae
1172092	Telomere Shortening	2376.166667	Oxidative Stress:AFFECTS:Telomere Shortening
597382	Endothelial dysfunction	2426.849057	Oxidative Stress:AFFECTS:Endothelial dysfunction
990601	cell homeostasis	2427.285714	Oxidative Stress:AFFECTS:cell homeostasis
1169253	Hepatocarcinogenesis	2428.388889	Oxidative Stress:AFFECTS:Hepatocarcinogenesis
1173169	Cell Death Process	2441.000000	Oxidative Stress:AFFECTS:Cell Death Process
1175118	Neuropathogenesis	2464.250000	Oxidative Stress:AFFECTS:Neuropathogenesis
990581	Induction of Apoptosis	2473.250000	Oxidative Stress:AFFECTS:Induction of Apoptosis
841628	Protein Expression	2480.846154	Oxidative Stress:AFFECTS:Protein Expression
1180739	cellular metabolism	2511.000000	Oxidative Stress:AFFECTS:cellular metabolism
1169594	Inflammation Process	2516.066667	Oxidative Stress:AFFECTS:Inflammation Process
594272	Perinatal brain damage	2533.333333	Oxidative Stress:AFFECTS:Perinatal brain damage
815529	Contraction	2534.750000	Oxidative Stress:AFFECTS:Contraction
950766	Degenerative disorder	2538.812500	Oxidative Stress:AFFECTS:Degenerative disorder
1197151	cell fate	2610.800000	Oxidative Stress:AFFECTS:cell fate
829564	chromatin remodeling	2613.750000	Oxidative Stress:AFFECTS:chromatin remodeling
831847	Base Excision Repair	2632.000000	Oxidative Stress:AFFECTS:Base Excision Repair

Figure 46 - Metabolic Diseases Semi-Metric Search List

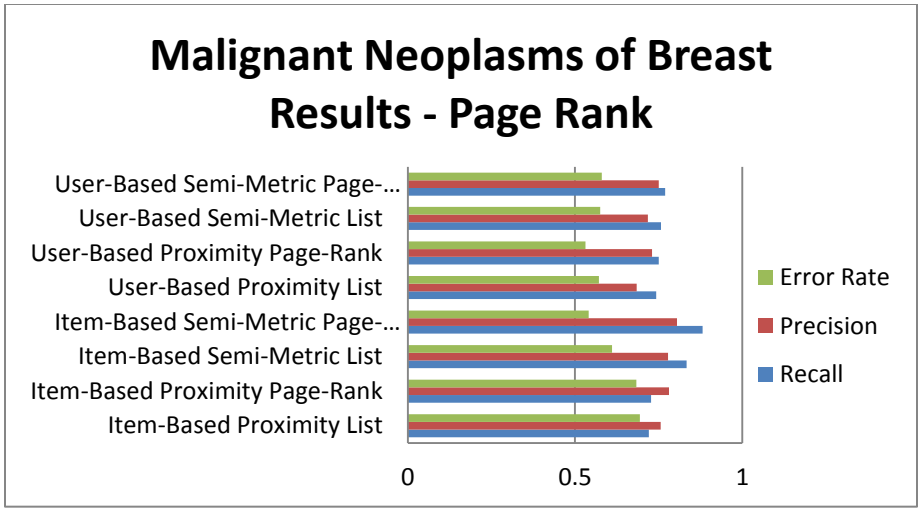


Figure 47 - Malignant Neoplasms of Breast Page Rank Bar Chart

Subject | Predication | Object

Submit

Searching for Malignant neoplasm of breast + AFFECTS + Mammary Tumorigenesis

	Concept	Distance	Path
16115	Patients	-1.952366	Malignant neoplasm of breast:OCCURS_IN:Patients
22886	Woman	-1.901194	Malignant neoplasm of breast:NEG_AFFECTS:Woman
10465	Human	-1.781454	Malignant neoplasm of breast:PART_OF:Human
3019	Malignant neoplasm of breast	0.000000	
14388	Neoplasms	0.085409	Malignant neoplasm of breast:OCCURS_IN:Neoplasms
13981	Mus	2.194712	Malignant neoplasm of breast:NEG_PROCESS_OF:Mus
18332	Rattus norvegicus	2.244216	Malignant neoplasm of breast:PART_OF:Rattus no...
6556	Canis familiaris	3.782288	Malignant neoplasm of breast:PART_OF:Canis fam...
13509	House mice	25.589744	Malignant neoplasm of breast:PART_OF:House mice
417680	Animal Model	36.000000	Malignant neoplasm of breast:PROCESS_OF:Animal...
1115659	Mammary Neoplasms	41.514286	Malignant neoplasm of breast:COEXISTS_WITH:Mam...
18960	Rodent	53.473684	Malignant neoplasm of breast:PROCESS_OF:Rodent
18336	Rats	129.363636	Malignant neoplasm of breast:PROCESS_OF:Rats
5638	Daughter	182.314815	Malignant neoplasm of breast:PROCESS_OF:Daughter
13526	Mice	187.940741	Malignant neoplasm of breast:PROCESS_OF:Mice
3364	Malignant Neoplasms	212.684956	Malignant neoplasm of breast:PRECEDES:Malignan...
13514	Mice	217.000000	Malignant neoplasm of breast:PART_OF:Mice
1169822	Mammary Tumorigenesis	227.800000	
1176425	Transgenic Model	244.800000	Malignant neoplasm of breast:AFFECTS:Transgeni...
1168139	Breast Cancer Model	340.000000	Malignant neoplasm of breast:PROCESS_OF:Breast...
18348	Rats	376.384615	Malignant neoplasm of breast:PROCESS_OF:Rats
12865	Mammary Tumor Virus	480.000000	Malignant neoplasm of breast:PROCESS_OF:Mammar...
1389	Animals	604.300000	Mammary Tumorigenesis:PROCESS_OF:Animals
13530	Mice	672.688889	Malignant neoplasm of breast:PART_OF:Mice
461834	Offspring	854.393939	Malignant neoplasm of breast:PROCESS_OF:Offspring
1391	Animals	922.307692	Malignant neoplasm of breast:PROCESS_OF:Animals
133604	Mice	934.800000	Mammary Tumorigenesis:PROCESS_OF:Mice
415175	mammary tumor virus	996.800000	Malignant neoplasm of breast:AFFECTS:Mammary T...

Figure 48 - Malignant Neoplasms of Breast Semi-Metric Search List

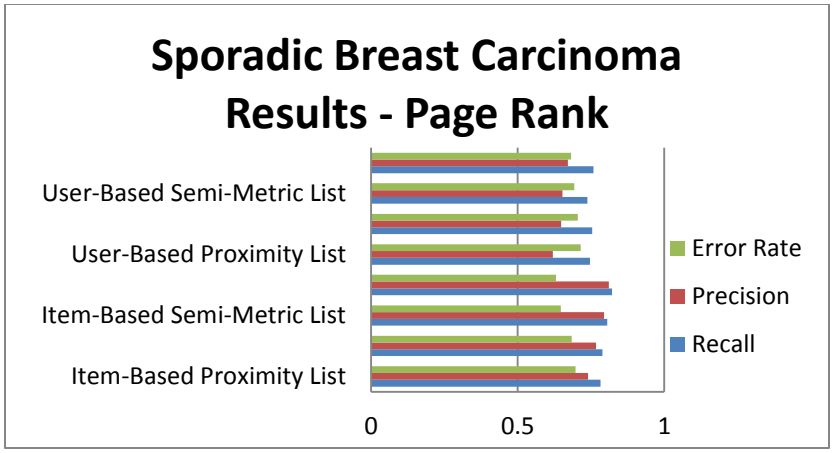


Figure 49 - Sporadic Breast Carcinoma Page Rank Bar Chart

Subject Predication Object

Submit

Searching for Sporadic Breast Carcinoma + AFFECTS + Gene-Environment Interaction

	Concept	Distance	Path
999811	Sporadic Breast Carcinoma	0.000000	
16115	Patients	6.327586	Sporadic Breast Carcinoma:PROCESS_OF:Patients
22886	Woman	15.541667	Sporadic Breast Carcinoma:PROCESS_OF:Woman
10465	Human	25.682927	Sporadic Breast Carcinoma:PROCESS_OF:Human
414933	Gene-Environment Interaction	451.500000	
129131	Individual	580.500000	Gene-Environment Interaction:PROCESS_OF:Indivi...
417680	Animal Model	610.000000	Gene-Environment Interaction:PROCESS_OF:Animal...
417765	trait	666.045455	Gene-Environment Interaction:AFFECTS:trait
13981	Mus	666.250000	Gene-Environment Interaction:PROCESS_OF:Mus
4067	Child Development	766.500000	Gene-Environment Interaction:COEXISTS_WITH:Chi...
126246	Antisocial behavior	789.250000	Gene-Environment Interaction:AFFECTS:Antisocia...
1389	Animals	815.500000	Gene-Environment Interaction:PROCESS_OF:Animals
2872	Body Size	842.500000	Gene-Environment Interaction:AFFECTS:Body Size
9021	Genotype	957.166667	Gene-Environment Interaction:AFFECTS:Genotype
13509	House mice	1037.000000	Gene-Environment Interaction:PROCESS_OF:House ...
266005	Obese build	1544.500000	Gene-Environment Interaction:AFFECTS:Obese build
8242	Fibrinolysis	2075.500000	Gene-Environment Interaction:AFFECTS:Fibrinolysis
22020	Twin Multiple Birth	2216.500000	Gene-Environment Interaction:PROCESS_OF:Twin M...
465585	high-risk group	2261.500000	Gene-Environment Interaction:PROCESS_OF:high-r...
6627	Drosophila melanogaster	2275.833333	Gene-Environment Interaction:PROCESS_OF:Drosop...
383950	At risk for suicide	2473.000000	Gene-Environment Interaction:AFFECTS:At risk f...
14128	Myopia	3961.000000	Gene-Environment Interaction:AFFECTS:Myopia
4525	Cleft Palate	4468.500000	Gene-Environment Interaction:AFFECTS:Cleft Palate
70946	biological adaptation to stress	5314.000000	Gene-Environment Interaction:AFFECTS:biologica...
992561	Abuse	5725.500000	Gene-Environment Interaction:AFFECTS:Abuse
71828	Premature Birth	5780.000000	Gene-Environment Interaction:AFFECTS:Premature...
54780	African American	6074.500000	Gene-Environment Interaction:PROCESS_OF:Africa...
17135	Polymorphism	6259.500000	Gene-Environment Interaction:AFFECTS:Polymorphism
460443	Plasticity	6867.000000	Gene-Environment Interaction:AFFECTS:Plasticity

Figure 50 - Sporadic Breast Carcinoma Semi-Metric Search List

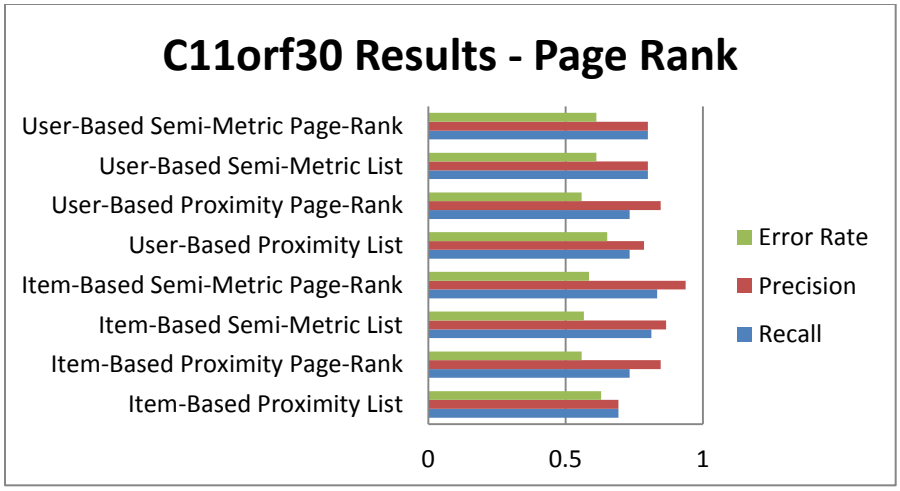


Figure 51 - c11orf30 Page Rank Bar Chart

Subject | Predication | Object

Submit

Searching for C11orf30 + ASSOCIATED_WITH + Malignant neoplasm of breast

	Concept	Distance	Path
1309066	C11orf30	0.000000	
999811	Sporadic Breast Carcinoma	298.000000	C11orf30:ASSOCIATED_WITH:Sporadic Breast Carci...
3019	Malignant neoplasm of breast	401.000000	
22886	Woman	404.633907	Malignant neoplasm of breast:AFFECTS:Woman
16115	Patients	410.763889	Malignant neoplasm of breast:AFFECTS:Patients
129131	Individual	473.076923	Malignant neoplasm of breast:AFFECTS:Individual
414653	Cell Proliferation	480.136364	Malignant neoplasm of breast:AFFECTS:Cell Prol...
80236	Apoptosis	597.363636	Malignant neoplasm of breast:AFFECTS:Apoptosis
1169822	Mammary Tumorigenesis	628.800000	Malignant neoplasm of breast:AFFECTS:Mammary T...
417680	Animal Model	652.333333	Malignant neoplasm of breast:AFFECTS:Animal Model
990998	mammary gland development	696.333333	Malignant neoplasm of breast:AFFECTS:mammary g...
1115659	Mammary Neoplasms	779.750000	Malignant neoplasm of breast:AFFECTS:Mammary N...
460820	gene polymorphism	839.285714	Malignant neoplasm of breast:AFFECTS:gene poly...
1176031	Second Degree Relative	898.333333	Malignant neoplasm of breast:AFFECTS:Second De...
17135	Polymorphism	923.304348	Malignant neoplasm of breast:AFFECTS:Polymorphism
608979	Counselees	1002.000000	Malignant neoplasm of breast:AFFECTS:Counselees
179114	Multiple alleles	1030.000000	Malignant neoplasm of breast:AFFECTS:Multiple ...
22420	Variation (Genetics)	1054.800000	Malignant neoplasm of breast:AFFECTS:Variation...
140787	Oncologist	1055.666667	Malignant neoplasm of breast:AFFECTS:Oncologist
200503	Sister	1056.700000	Malignant neoplasm of breast:AFFECTS:Sister
466302	Secondary malignant neoplasm of axilla	1088.666667	Malignant neoplasm of breast:AFFECTS:Secondary...
1176484	Tumor-Associated Process	1124.500000	Malignant neoplasm of breast:AFFECTS:Tumor-Ass...
126074	Term Birth	1151.000000	Malignant neoplasm of breast:AFFECTS:Term Birth
114763	Malignant pericardial effusion	1154.500000	Malignant neoplasm of breast:AFFECTS:Malignant...
111225	Germ-Line Mutation	1173.000000	Malignant neoplasm of breast:AFFECTS:Germ-Line...
951687	Dietary intake	1268.428571	Malignant neoplasm of breast:AFFECTS:Dietary i...
460569	Dormancy	1272.500000	Malignant neoplasm of breast:AFFECTS:Dormancy

Figure 52 - c11orf30 Semi-Metric Search List

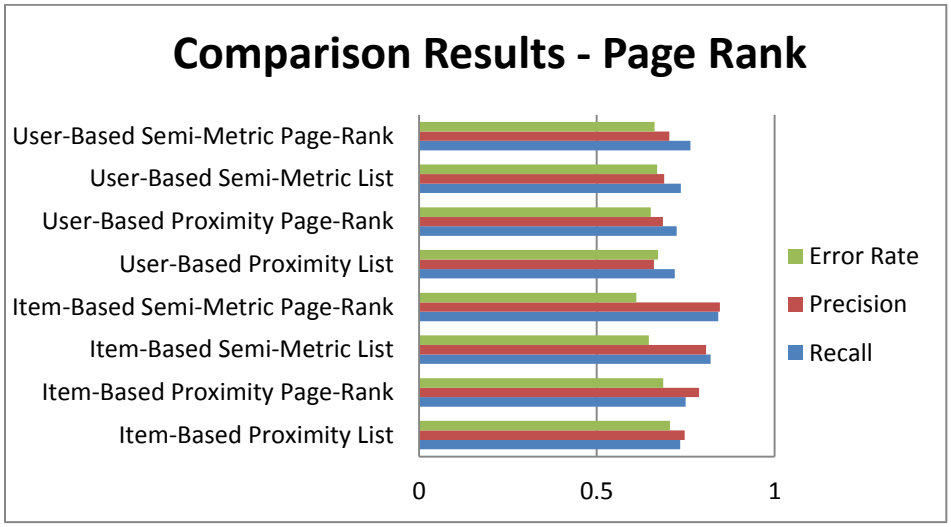


Figure 53 - Overall Page Rank Results Bar Chart

Combined Results all Medical Queries – Page Ranking

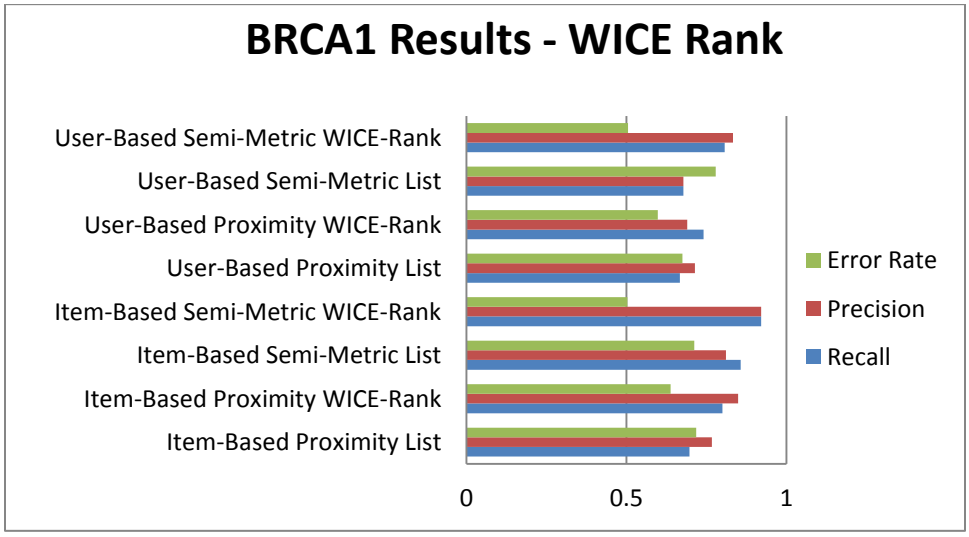


Figure 54 - BRCA1 WICE Bar Chart

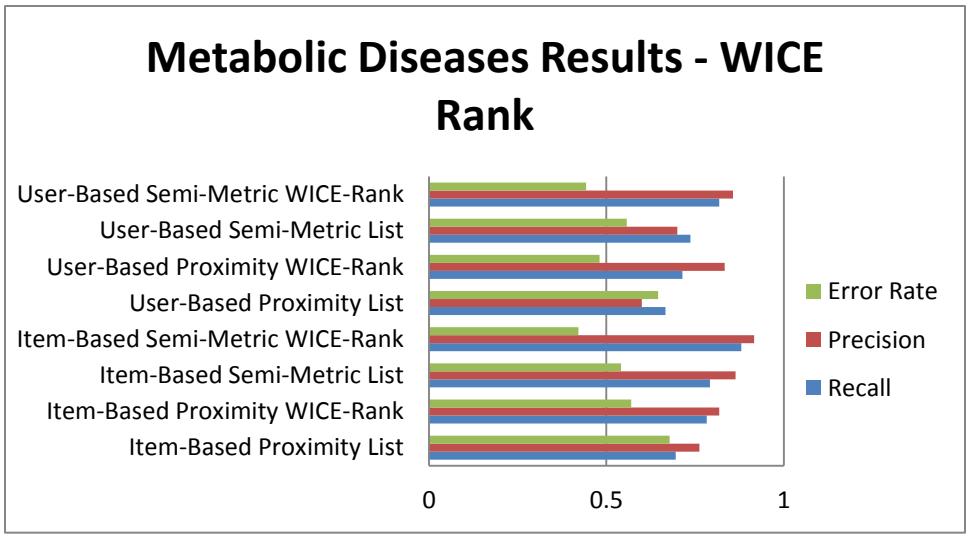


Figure 55 - Metabolic Diseases WICE Bar Chart

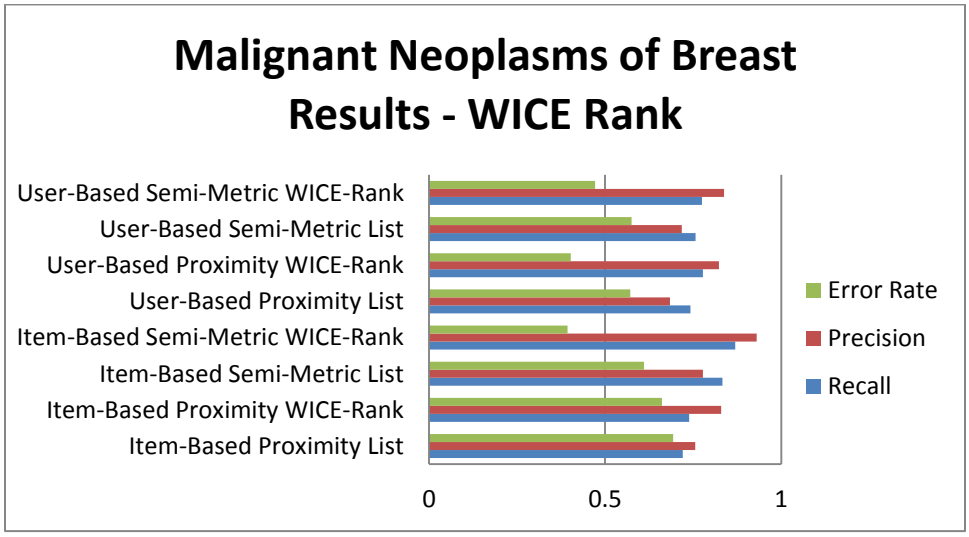


Figure 56 - Malignant Neoplasms of Breast WICE Bar Chart

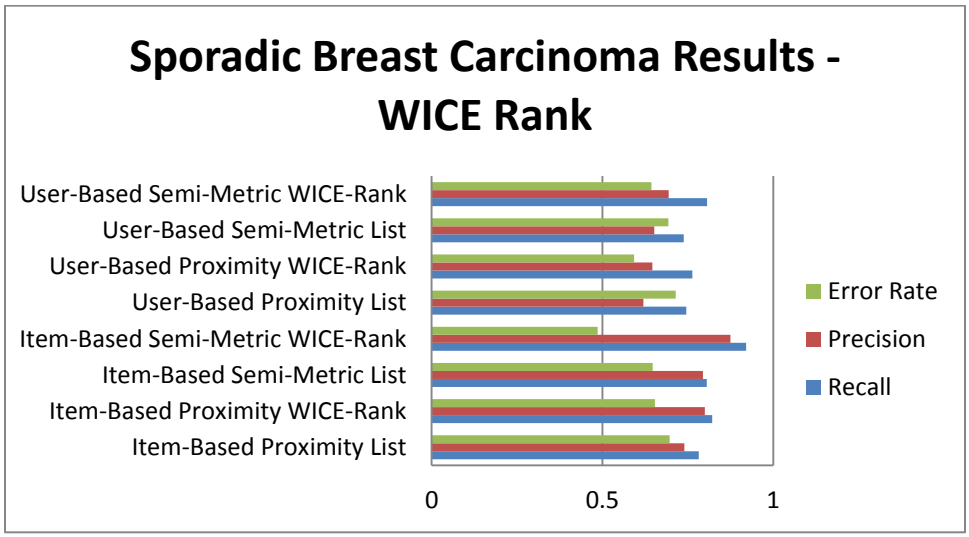


Figure 57 - Sporadic Breast Carcinoma WICE Bar Chart

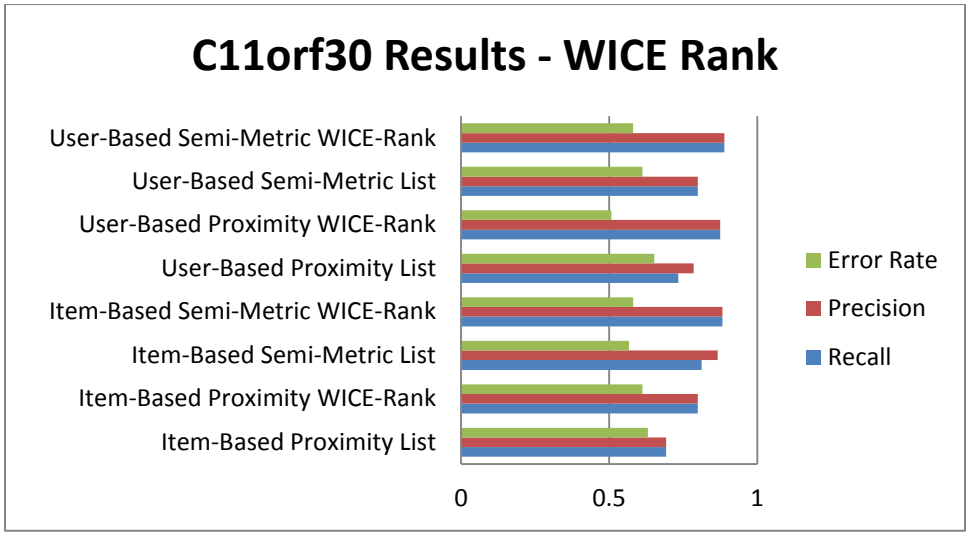


Figure 58 - c11orf30 WICE Bar Chart

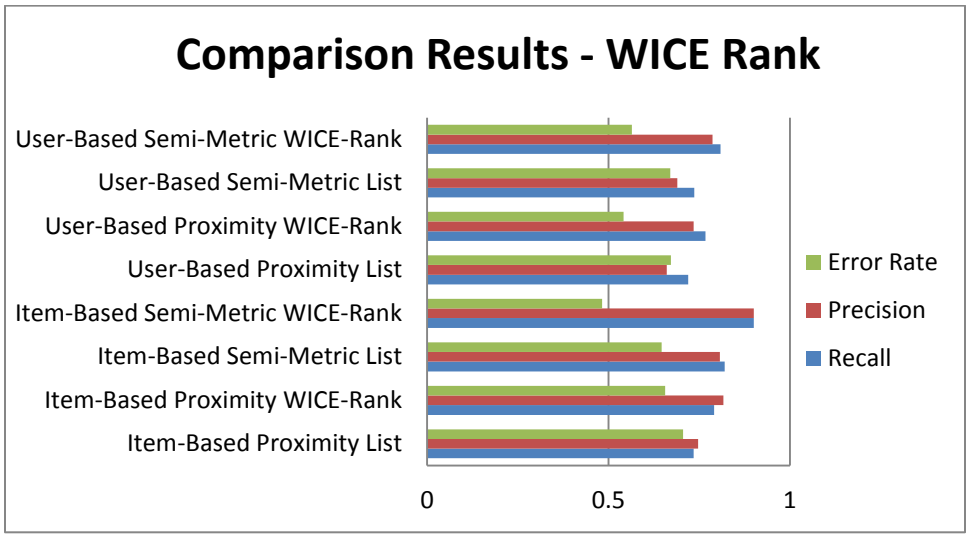


Figure 59 - Combined WICE Results Bar Chart

Combined Results all Medical Queries – WICE Ranking

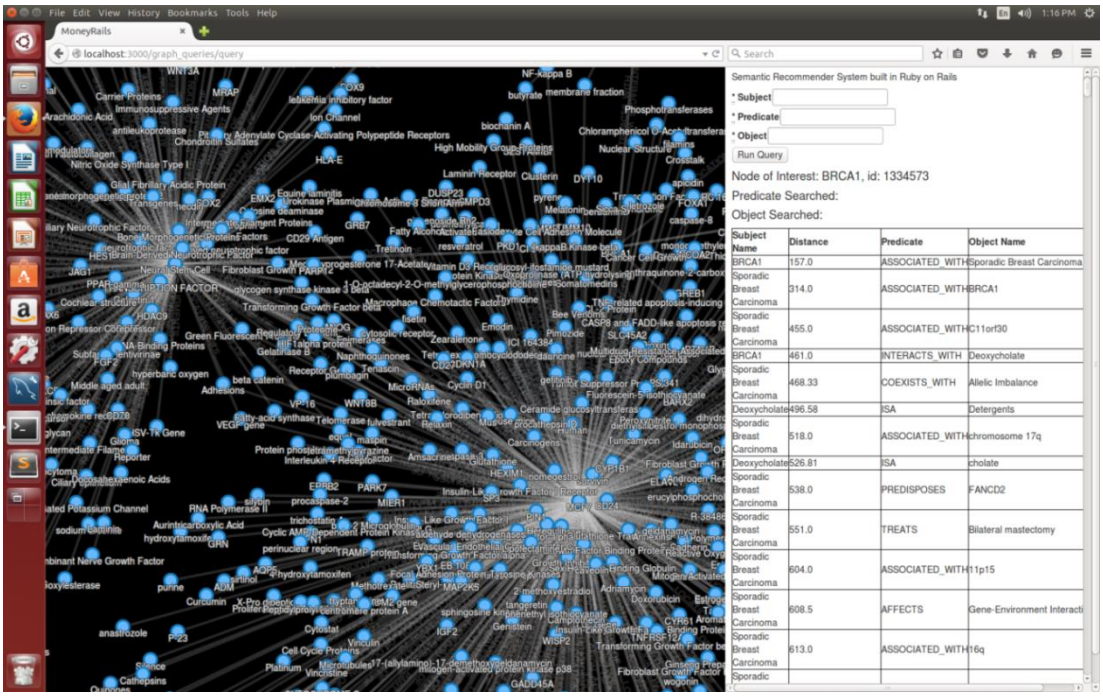


Figure 60 - BRCA1 Example Visualization in Ruby Environment

REFERENCES

[0] L. Kerschberg and H. Jeong, "Ubiquitous Data Management in Knowledge Sifter via Data-DNA," Ubiquitous Data Management, 2005.

[1] L. Zhen, H.-T. Song, and J.-T. He, "Recommender systems for personal knowledge management in collaborative environments," *Expert Systems with Applications: An International Journal*, vol. 39, no. 16, Nov. 2012.

[2] P. Dwivedi and K. K. Bharadwaj, "e-learning recommender system for learners in online social networks through association retrieval," presented at the CUBE '12: Proceedings of the CUBE International Information Technology Conference, 2012.

[3] A. Vakali, "Evolving social data mining and affective analysis methodologies, framework and applications," presented at the IDEAS '12: Proceedings of the 16th International Database Engineering & Applications Sysmposium, 2012.

[4] J. Gemmell, T. Schimoler, B. Mobasher, and R. Burke, "Resource recommendation in social annotation systems: A linear-weighted hybrid approach," *Journal of Computer and System Sciences*, vol. 78, no. 4, Jul. 2012.

[5] H. Zhang, Y. Gao, H. Chen, and Y.-S. Li, "TravelHub: A semantics-based mobile recommender for composite services," presented at the Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on, 2012, pp. 476–482.

[6] Y. Zhang, Y. Zheng, and J. Ni, "Context-Aware Commodity Recommendation Information Service in Ecommerce," presented at the Internet Computing for Science and Engineering (ICICSE), 2012 Sixth International Conference on, 2012, pp. 20–25.

[7] M. E. Alper and S. G. O x0308 gu x0308 du x0308 cu x0308, "Personalized Recommendation in Folksonomies Using a Joint Probabilistic Model of Users, Resources and Tags," presented at the Machine Learning and

Applications (ICMLA), 2012 11th International Conference on, 2012, vol. 1, pp. 368–373.

[8] V. Grou x00E8 s, Y. Naudet, and O. Kao, "Combining Linguistic Values and Semantics to Represent User Preferences," *Semantic Media Adaptation and Personalization (SMAP), 2011 Sixth International Workshop on*, pp. 27–32, 2011.

[9] H. Wermser, A. Rettinger, and V. Tresp, "Modeling and Learning Context-Aware Recommendation Scenarios Using Tensor Decomposition," presented at the Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on, 2011, pp. 137–144.

[10] G. Adomavicius and A. Tuzhilin, "Context-Aware Recommender Systems," in *Recommender Systems Handbook*, no. 7, Boston, MA: Springer US, 2010, pp. 217–253.

[11] R. Wetzker, C. Zimmermann, and C. Bauckhage, "Detecting Trends in Social Bookmarking Systems: A del.icio.us Endeavor," *International Journal of Data Warehousing and Mining*, vol. 6, no. 1, Jan. 2010.

[12] I. Torre, "Adaptive systems in the era of the semantic and social web, a survey," *User Modeling and User-Adapted Interaction*, vol. 19, no. 5, pp. 433–486, Nov. 2009.

[13] A. García-Crespo, J. Chamizo, I. Rivera, M. Mencke, R. Colomo-Palacios, and J. M. Gómez-Berbís, "SPETA: Social pervasive e-Tourism advisor," *Telematics and Informatics*, vol. 26, no. 3, pp. 306–315, Aug. 2009.

[14] L. Zhuhadar and O. Nasraoui, "Personalized cluster-based semantically enriched web search for e-learning," *ONISW '08: Proceeding of the 2nd international workshop on Ontologies and information systems for the semantic web*, Oct. 2008.

[15] Y. Naudet, S. Mignon, L. Lecaque, C. Hazotte, and V. Grou x00E8 s, "Ontology-Based Matchmaking Approach for Context-Aware Recommendations," presented at the Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS '08. International Conference on, 2008, pp. 218–223.

[16] I. X. N. Cantador, A. Bellogin, and P. Castells, "Ontology-Based Personalized and Context-Aware Recommendations of News Items," presented at the Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on, 2008, vol. 1, pp. 562–565.

[17] P. Brézillon, "From expert systems to context-based intelligent assistant systems: a testimony," *The Knowledge Engineering Review,* vol. 26, no. 1, pp. 19–24, Feb. 2011.

[18] M. Zacarias, H. S. Pinto, and J. Tribolet, "Integrating Engineering, Cognitive and Social Approaches for a Comprehensive Modeling of Organizational Agents and Their Contexts," *ContEXT'07: Proceedings of the 6th international and interdisciplinary conference on Modeling and using context,* vol. 4635, pp. 517–530, 2007.

[19] A. Zimmermann, A. Lorenz, and R. Oppermann, "An Operational Definition of Context," *ContEXT'07: Proceedings of the 6th international and interdisciplinary conference on Modeling and using context,* vol. 4635, pp. 558–571, 2007.

[20] Z. Yu, Y. Nakamura, S. Jang, S. Kajita, and K. Mase, "Ontology-Based Semantic Recommendation for Context-Aware E-Learning," in *Ubiquitous Intelligence and …*, vol. 4611, no. 88, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 898–907.

[21] C. R x00E4 ck, S. Arbanowski, and S. Steglich, "Context-aware, ontology-based recommendations," *Applications and the Internet Workshops, 2006. SAINT Workshops 2006. International Symposium on*, p. 7, 2006.

[22] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the stateof-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp.734–749, Jun. 2005.

[23] B. Mrohs, M. Luther, R. Vaidya, and M. Wagner, "OWL-SF–a distributed semantic service framework," *Proc. Of the Workshop on Context Awareness for …*, 2005.

[24] M. Bazire and P. Brézillon, "Unerstanding Context Before Using It," *CONTEXT 2005,* vol. 3554, pp. 29–40, 2005.

[25] J. O'Donovan and B. Smyth, "Trust in recommender systems," presented at the the 10th international conference, New York, New York, USA, 2005, p. 167.

[26] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 89–115, Jan. 2004.

[27] R. Kaschek, K.-D. Schewe, B. Thalheim, and L. Zhang, "Integrating Context in Modeling for Web Information Systems," *Integrating Context in Modelling for Web Information Systems*, vol. 3095, no. LNCS, pp. 77–88, 2004.

[28] - Tsvi Kuflik Shlomo Berkovsky, Francesca Carmagnola Dominikus Heckmann, Antonio Krüger (Eds.), "Advances in Ubiquitous User Modelling", University of Haifa

[29] –Wei Shen, Jianyong Wang, Ping Luo, Min Wang, "A Graph-Based Approach for Ontology Population with Named Entities", Proceedings of the 21st ACM international conference on Information and knowledge management, Pages 345-354

[30] – Tsvi Kuflik, Shlomo Berkovsky, Francesca Carmagnola, Dominikus Heckmann, Antonio Krüger, "Advances in Ubiquitous User Modeling", University of Haifa

[31] – Wei Shen, Jianyong Wang, Ping Luo, Min Wang, "APOLLO: A General Framework for Populating Ontology with Named Entities via Random Walks on Graphs", Proceedings of the 21st International Conference on World Wide Web, Pages 595-596

[32] – Shengbo Guo, "Bayesian Recommender Systems: Models and Algorithms", The Australian National University

[33] – Gabor Magyar and Gabor Knapp, "Advances in Information Systems Development: New Methods and Practice", Springer book, Budapest University of Technology

[34] - Ricardo Couto, Antunes da Rocha, Markus Endler, "Context Management for Distributed and Dynamic Context-Aware Computing for the Networked Society", Institute of Informatics, Federal University of Goias

[35] – Jeremy Debattista, Simon Scerri, Ismael Rivera, and Siegfried Handschuh, "Ontology-based Rules for Recommender Systems", Digital Enterprise Research Institute, National University of Ireland,

[36] – Martin Atzmueller, Alvin Chin, Denis Helic, Andreas Hotho, "Modeling and Mining Ubiquitous Social Media", International Workshops MSM 2011, Boston, MA

[37] – Feng Gao, Sami Bhiri, and Zhangbing Zhou, "User-centric modeling and processing for ubiquitous events using semantic capability models", Communications in Mobile Computing, December 2012, 1:7

[38] – Liliana Ardissono, Paul Brna, Antonija Mitrovic, "User Modeling 2005", 10th International Conference, UM 2005, Edinburgh, Scotland, UK, July 24-29, 2005.

[39] – Abdulbaki Uzun, Christian Rack, "Using a Semantic Multidimensional Approach to Create a Contextual Recommender System", from psu.edu

[40] – Dominik Heckmann, and Antonio Krueger, "A User Modeling Markup Language (UserML) for Ubiquitous Computing", 9th International Conference, UM 2003 Johnstown, PA, USA, June 22–26, 2003 Proceedings

[41] – "A Multi-Purpose Ontology-Based Approach for Personalised Content Filtering and Retrieval", Ivan Cantador, Miriam Fernandez, David Vallet, Pablo Castells, J´erome Picault, and Myriam Ribiere, Advances in Semantic Media Adaptation and Personalization, Volume 93 of the series Studies in Computational Intelligence pp 25-51

[42] - "A survey and classification of semantic search approaches", Christoph Mangold, International Journal of Metadata, Semantics and Ontologies

[43] – "Automatic ontology based User Profile Learning from heterogeneous Web Resources in a Big Data Context", Anett Hoppe, Journal Proceedings of the VLDB Endowment, Volume 6 Issue 12, August 2013, Pages 1428-1433

[44] – "Capturing Interest Through Inference and Visualization: Ontological User Profiling in Recommender Systems", Stuart E. Middleton, Nigel R. Shadbolt, David C. De Roure, K-CAP '03 Proceedings of the 2nd international conference on Knowledge capture, pages 62-69, ACM, New York

[45] – "Concept based Personalized Web Search", S. Sendhilkumar* and T. V. Geetha*, Norwegian University of Science and Technology

[46] – "Contextual Semantic Search Navigation", Geir Solskinnsbakk

[47] – "DISCOVERING THE IMPACT OF KNOWLEDGE IN RECOMMENDER SYSTEMS: A COMPARATIVE STUDY", Bahram Amini, Roliana Ibrahim, Mohd Shahizan Othman, Cornell University Library

[48] – "Emerging Graph Queries In Linked Data", Arijit Khan, Yinghui Wu, Xifeng Yan, Dept. of Comput. Sci., Univ. of California, Santa Barbara, CA, USA

[49] – "Exploiting Synergy Between Ontologies and Recommender Systems", Stuart E. Middleton, Harith Alani, David C. De Roure, Cornell University Library

[50] – "Next generation knowledge access", John Davies, Alistair Duke, Nick Kings, Journal of Knowledge Management

[51] – "Ontological Approach in Knowledge Based Recommender System to Develop the Quality of E-learning System", Saman Shishehchi, Seyed Yashar Banihashem, Nor Azan Mat Zin, Shahrul Azman Mohd. Noah, Australian Journal of Basic and Applied Sciences 02/2012; 6(2):115-123

[52] – "Ontological User Profiling in Recommender Systems", STUART E. MIDDLETON, NIGEL R. SHADBOLT, and DAVID C. DE ROURE, Intelligence, Agents, Multimedia Group, University of Southampton

[53]- "Ontology-based Multimedia Contents Retrieval Framework in Smart TV Environment", Moohun LEE*, Joonmyun CHO*, Jeongju Yoo*, Jinwoo Hong, Next Generation SmartTV Research Department, ETRI(Electronics and Telecommunications Research Institute), Korea

[54] – "ONTOLOGY-DRIVEN PERSONALIZED QUERY REFINEMENT", SOFIA STAMOU, LEFTERIS KOZANIDIS, PARASKEVI TZEKOU, NIKOS ZOTOS, Computer Engineering and Informatics Department, Patras University, Journal of Web Engineering

[55] – "PERSONALIZED ONTOLOGY BASED CONTEXT AWARE RECOMMENDER SYSTEM", NUPUR GIRI, VIDYA ZOPE, HOD of Computer Engineering Department, VESIT, Chembur, Mumbai, India

[56] – "Semantic Association Analysis in Ontology-based Information Retrieval", Payam M. Barnaghi, Handbook of Research on Digital Libraries: Design, Development, and Impact, pp141

[57] – "Semantic Web Personalization: A Survey", Ayesha Ameen Khaleel Ur Rahman Khan B.Padmaja Rani, Information and Knowledge Management

[58] – "SharePoint Insight: Semantic Social Search on Enterprise 2.0 Portals", Pascal van Delft, University of Amsterdam

[59] – "Using a concept-based user context for search personalization", Mariam Daoud, Lynda Tamine-Lechani, Mohand Boughanem, Proceedings of the World Congress on Engineering 2008 Vol I, WCE 2008, July 2 - 4, 2008,

[60] – "Web Search Personalization with Ontological User Profiles", Ahu Sieg, Center for Web Intelligence, School of Computer Science, Telecommunication and Information Systems, DePaul University

[61] – "Semantic Medline: A Web Application for Managing the Results of PubMed Searches", Fiszman, Rodriguez, Shin, et al, National Library of Medicine

[62] – "Semi-metric Behavior in Document Networks and its Application to Recommendation Systems", Rocha, IOS Press, pp 137-163

[63] – "Semi-metric networks for Recommender Systems", Simas, Rocha, Cognitive Science Program, Indiana University

[64] – "Mining Indirect Association Rules for Web Recommendation", Kazienko, Institute of Informatics, Wrocław University of Technology

[65] –"Semantic Medline: A Web Application for Managing the Results of PubMed Searches", Kilicoglu, Fiszman, et. Al, National Library of Medicine

[66] – "Graph Database Applications and Concepts with Neo4J", Miller, Georgia Southern University

[67] – "Evaluating Recommendation Systems", Shani and Gunawardana, Microsoft Research,

[68] – "Using proximity to compute semantic relatedness in RDF graphs", Leal, University of Porto, Portugal

[69] – "Distance Closures on Complex Networks" – Simas, Rocha, 2014, 1Cognitive Science Program, Indiana University,

[70] – "Automatic classification using supervised learning in a medical document filtering application", Mostafaa, Lamb, Information Processing & Management, Volume 36, Issue 3, 1 May 2000, Pages 415–444

[71] – "Classification of Web Documents Using a Graph Model", Schenker, Last, Bunke, Kandel, University of South Florida, Department of Computer Science and Engineering

[72] – "Algorithms for Graph Similarity and Subgraph Matching", Koutra, Parikh, Ramdas, and Xiang, Computer Science Department, Carnegie-Mellon University

[73] - "SimRank: A measure of Structured-Context Similarity", Jeh, Widom, Stanford University

[74] – "Graph Classification and Clustering based on Vector Space Embedding", Riesen, Bunke, World Scientific Publishing Company

[75] – "Graph-Based Clustering and Data Visualization Algorithms", Vathy-Fogarassy, Abonyi, Springer Briefs in Computer Science

[76] – "WICER: A Weighted Inter-Cluster Edge Ranking for Clustered Graphs", Padmanabhan, Desikan, Srivastava, Riaz, Dept. of Computer Science University of Minnesota

[77] - "Link Prediction Algorithms", http://be.amazd.com/link-prediction/

[78] - "Google Page Rank", http://pr.efactory.de/e-pagerank-algorithm.shtml

[79] – "Recommender Systems: Collaborative Filtering and Other Methods", Xavier Amatriain, Netflix

[80] – "The Interest Graph Maps Our Connections to Ideas and Things", Gideon Rosenblatt , http://www.the-vital-edge.com/shared-interest-graph-in-work/

BIOGRAPHY

Gary G. Roberson graduated from Wheaton High School, Wheaton, Maryland and born at Georgetown Hospital in Washington DC. His family lives mostly in Montgomery and Frederick counties in Maryland. He received his Bachelor of Science from Carnegie Mellon University with a major in Electrical Engineering graduating magna cum laude and accepted into Phi Kappa Phi, Tau Betta Pi, and Eta Kappa Nu all academic honorary societies. He was fraternity President for Sigma Alpha Epsilon while at CMU. He received his Master of Science in Computer Science from Johns Hopkins University. He received his Master of Business Administration from George Washington University in Marketing. He received his Master in Computational Science and Informatics from George Mason University in 2013. He has worked for the Department of the Navy, Naval Surface Weapons Systems Command working on strategic systems as an engineer, the International Trade Administration at the Department of Commerce implementing the Worldwide Trade and Information System (WITS), and as a manager for the AES Corporation headquartered in Arlington, Virginia for global enterprise software. He has also worked as an Enterprise Software consultant and consulting services entrepreneur for the Adapted Enterprise Technology LLC (www.iadapted.com) including projects with many companies and government agencies. Some of these enterprises include Ernst and Young, the National Association of Broadcasters, the Homeland Security Administration, Veterans Affairs, the Federal Reserve Board, Maryland Department of Education, and a large number of smaller and middle sized enterprises mostly in the mid-Atlantic region. His specialty areas are in Enterprise Resource Planning, Customer Relationship Management, Enterprise Analytics, Data Science, and Enterprise and Solution Architecture for application and data systems design. In the future, he hopes to have the opportunity to teach students some of the knowledge he has accumulated in these topics.