I, Ahmad Rawashdeh, hereby submit this original work as part of the requirements for the degree of Doctor of Philosophy in Computer Science & Engineering.

It is entitled:
Semantic Similarity of Node profiles in Social Networks

Student's name: Ahmad Rawashdeh

This work and its defense approved by:

Committee chair: Anca Ralescu, Ph.D.
Committee member: Irene Diaz, Ph.D.
Committee member: Rehab M Duwairi, Ph.D.
Committee member: Kenneth Berman, Ph.D.
Committee member: Chia Han, Ph.D.
Committee member: Dan Ralescu, Ph.D.
Semantic Similarity of Node Profiles in Social Networks

A Dissertation Submitted to the

Graduate School
of the University of Cincinnati

in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

In the Department of
Electrical Engineering and Computing Systems
of College of Engineering and Applied Sciences

July 2015

By

Ahmad Y. Rawashdeh

M.S. Computer Science, Jordan University of Science and Technology, Jordan, January 2009

B.S. Computer Science, Jordan University of Science and Technology, Jordan, June 2006
Abstract

It can be said, without exaggeration, that social networks have taken a large segment of population by a storm. Regardless of the actual geographical location, of socio-economic status, as long as access to an internet connected computer is available, a person has access to the whole world, and to a multitude of social networks. By being able to share, comment, and post on various social networks sites, a user of social networks becomes a “citizen of the world”, ensuring presence across boundaries (be they geographic, or socio-economic boundaries).

At the same time social networks have brought forward many issues interesting from computing point of view. One of these issue is that of evaluating similarity between nodes/profiles in a social network. Such evaluation is not only interesting, but important, as the similarity underlies the formation of communities (in real life or on the web), of acquisition of friends (in real life and on the web).

In this thesis, several methods for finding similarity, including semantic similarity, are investigated, and a new approach, Wordnet-Cosine similarity is proposed. The Wordnet-Cosine similarity (and associated distance measure) combines both a lexical database, Wordnet, with Cosine similarity (from information retrieval) to find possible similar profiles in a network.

In order to assess the performance of Wordnet-Cosine similarity measure, two experiments have been conducted. The first experiment illustrates the use for Wordnet-Cosine similarity in community formation. Communities are considered to be clusters of profiles. The results of using Wordnet-Cosine are compared with those using four other similarity measures (also described in this thesis). In the second set of experiments, Wordnet-Cosine was applied to the problem of link prediction. Its performance of predicting links in a random social graph was compared with a random link predictor and was found to achieve better accuracy.
Acknowledgements

This thesis marks the closing chapter of my study at the University of Cincinnati, where I spent a wonderful time during which I learned a lot about research and teaching.

First, I would like to thank very much my adviser, Prof. Anca Ralescu, for her continuous effort in advising me and guiding me through my Ph.D journey. She has taught me and helped me understand difficult issues through numerous individual and group meetings. She never ceases to surprise me with her gentle soul and her kind words. Her dedication to her students is something that I can’t forget. I strongly recommend her to any student who is eager to learn and be inspired to pursue her/his path through the field of Computer Science.

Thanks are due to the members of the PhD committee professors Ken Berman, Irene Diaz Rodriguez, Rehab Duwairi, Chia Han, and Dan Ralescu for their agreement to be in my Ph.D thesis committee, and valuable comments they made on the occasion of my proposal defense. These comments guided my research at subsequent stages.

I gained valuable teaching experience as a Teaching Assistant for various Computer Science courses. I was fortunate to work with the amazing professor Berman, professor Cheng, and again, with my adviser, professor Ralescu.

Mrs. Julie Muenchen of Graduate Studies Office helped me with the procedures and regulations toward completing all the requirements and for this I thank her too. I would like to thank my host family Mrs. and Mr. Mary and Larry Collins. They welcomed me in their home and introduced us to the wonderful American traditions. I made many friends among my group mates and other students at the University of Cincinnati. I’m grateful to all for their friendship.

There is no better place and time to thank again my family. I’m grateful to my parents, my mother Jamelah and my father Yousef, for their love and unconditional support. My dream of achieving this moment in my studies became their dream.
My heartfelt thanks and love go to my siblings, my elder brother, Mohammad the philosopher, who preceded me in earning his PhD in Computer Science at University of Cincinnati, my lovely and intelligent sister, Rabeah, PhD in nanotechnology, who has supported me for as long as I can remember, and my younger brothers, Omar and Amer, biologist and mathematician respectively, who bring joy and cheer us all.
# Contents

1 Introduction ................................................. 1
   1.1 The Similarity Problem in Social Network ................. 1
       1.1.1 Social Network and background .................... 1
       Account .................................................. 2
       Profile ............................................... 2
       Friends ................................................. 2
       Search ................................................ 3
       Messages, Notifications, and Special Features .......... 3
       Limitations and Fellowship vs Friendship ............. 4
   1.1.2 Similarity ........................................... 4
   1.2 Organization of the Thesis ............................. 5

2 Initial Experiment .......................................... 6
   2.1 Initial Experiment .................................... 6
       2.1.1 Related Work on Similarity ...................... 6
       2.1.2 Datasets ........................................... 10
           Facebook Dataset ................................... 10
           The authorship dataset (DBLP) .................... 11
       2.1.3 Initial System Architecture ...................... 11
       2.1.4 Results of Initial Experiments ................. 12
       2.1.5 Conclusion for Chapter 2 ....................... 12
### 3 Similarity Between Nodes in a Social Network

3.1 Similarity in Social Networks

3.2 Semantic Similarity

3.3 Problem description and evaluation metrics

3.4 Motivation for finding similarity

3.5 Node and other similarity measurements
   - 3.5.1 Node Similarity
   - 3.5.2 Edge similarity
   - 3.5.3 Global Structural Similarities
   - 3.5.4 Conclusion for Chapter 3

### 4 Semantic Similarity Measures

4.1 Similarity measures and Semantic

4.2 Finding Similar Profiles
   - 4.2.1 Wordnet
     - Hypernym of a word
     - Wordnet Java API
   - 4.2.2 Semantic Tagger
   - 4.2.3 The occurrence frequency similarity(OF)

4.3 A unified Similarity Measure: Wordnet-cosine similarity

4.4 Experimental Results
   - 4.4.1 Facebook profiles data-set
   - 4.4.2 Results

4.5 Conclusion for Chapter 4

### 5 Algorithms and Implementations for Finding Similarity

5.1 Java algorithms for finding similarities
   - 5.1.1 Time Complexity Analysis of the Similarity Algorithms
   - 5.1.2 Space complexity of the similarity algorithms
6 Experiments and Results

6.1 Experiments

6.2 Creating the similarity and Distance Matrices

6.3 Clustering using Similarity and the Silhouette Index for Clustering Validation

6.3.1 Kmeans Clustering

6.3.2 Silhouette

6.3.3 Facebook

Set Distance
Wordnet-Cosine Distance
Semantic Categories Distance
Word Frequency Vector Distance
Average Silhouette

6.3.4 DBLP

Set Distance
Wordnet-Cosine Distance
Semantic Categories Distance
Word Frequency Vector Distance
Average Silhouette

6.3.5 Movielens

Set Distance
Wordnet-Cosine Distance
Semantic Categories Distance
Word Frequency Vector Distance
Average Silhouette

6.4 Comparison between Different Measures

6.5 Correlation between the Similarity Measures

6.6 Link Prediction using the Wordnet-Cosine similarity/distance measure

6.6.1 Wordnet-Cosine and Random Graph Adjacency

6.7 Conclusion to Chapter 6
8 Appendix

8.1 Code

8.1.1 Clustering Code

8.1.2 Avg Silhouette Code

8.1.3 Modified Kmeans (KernelKmean) Code

8.1.4 Link Prediction Experiment

8.2 System specification
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Small network with node profiles expressed as strings of characters.</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>Usage - Link prediction on the network illustrated in Figure 2.1 when node profile similarities are computed with the Jaccard index.</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Frequencies of the top 20 Movies interest in the Facebook dataset.</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>Architecture</td>
<td>12</td>
</tr>
<tr>
<td>2.5</td>
<td>Results (Similarity Histograms) - Facebook dataset</td>
<td>13</td>
</tr>
<tr>
<td>2.6</td>
<td>Results (Similarity Histogram) - DBLP dataset</td>
<td>13</td>
</tr>
<tr>
<td>4.1</td>
<td>Histogram of OF and WordNet similarity using the Facebook dataset.</td>
<td>34</td>
</tr>
<tr>
<td>4.2</td>
<td>Histogram of the four similarity measures using the Facebook dataset</td>
<td>34</td>
</tr>
<tr>
<td>5.1</td>
<td>Dependency between the procedures used to find the similarity measure</td>
<td>38</td>
</tr>
<tr>
<td>5.2</td>
<td>Dependency between the procedures used to find the Word Frequency Vector measure</td>
<td>38</td>
</tr>
<tr>
<td>6.1</td>
<td>Histogram of distances of Facebook data for the four similarity measures</td>
<td>51</td>
</tr>
<tr>
<td>6.2</td>
<td>Histogram of Distances of MovieLens data for the four Similarity Measures</td>
<td>52</td>
</tr>
<tr>
<td>6.3</td>
<td>Histogram of Distances of DBLP data for the four Similarity Measures</td>
<td>53</td>
</tr>
<tr>
<td>6.4</td>
<td>The Image Plots of Facebook Distance Matrices</td>
<td>54</td>
</tr>
<tr>
<td>6.5</td>
<td>Clustering using Set Distance applied on Facebook Dataset</td>
<td>59</td>
</tr>
<tr>
<td>6.6</td>
<td>Clustering using WordNet-Cosine Distance applied on Facebook Dataset</td>
<td>59</td>
</tr>
<tr>
<td>6.7</td>
<td>Clustering using Semantic Categories Distance applied on Facebook Dataset</td>
<td>59</td>
</tr>
<tr>
<td>6.8</td>
<td>Clustering using Word Frequency Vector Distance applied on Facebook Dataset</td>
<td>60</td>
</tr>
<tr>
<td>6.9</td>
<td>Average Silhouette for clustering the Facebook Dataset into 1-6 Clusters using different Distance Measures</td>
<td>60</td>
</tr>
</tbody>
</table>
# List of Tables

1.1 Comparison between different social networks .............................. 4

2.1 Pairwise similarity values for the network in Figure [2.1] ................... 9

2.2 Statistics of the Facebook Dataset ............................................. 11

3.1 Two Facebook profiles .............................................................. 20

3.2 Node and Link similarities ......................................................... 22

3.3 Comparison between similarity measures ..................................... 24

3.4 Comparison between similarity measures ..................................... 25


4.2 The cosine similarity of vectors $v$ .............................................. 31

4.3 Illustration of Movie Attribute of Facebook profiles: their tags and Hypernyms ... 32

4.4 OF and Wordnet-Cosine similarity of two Facebook profiles along their Movie Attribute. Profile IDs are partially masked for privacy ................................................. 32

4.5 User rating of the similarity between profile 1 and profile 2 using Facebook dataset in table 4.4 ................................................................. 33

4.6 Genres for movies listed in Table 4.4 ........................................... 33

5.1 Running time complexity of the similarity algorithms ..................... 48

6.1 The Format for the Movielens Dataset ........................................ 50

6.2 Facebook Dataset: the First 10 Records ...................................... 50

6.13 Accuracy of Wordnet-Cosine and Random Predictors .................. 70

6.3 MovieLens Dataset: the First 10 Records ..................................... 79
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4</td>
<td>DBLP Dataset: the First 10 Records</td>
<td>79</td>
</tr>
<tr>
<td>6.5</td>
<td>Characteristic of Distance Matrices used for Clustering</td>
<td>80</td>
</tr>
<tr>
<td>6.6</td>
<td>Distance Data Matrices from the Facebook Dataset (FD)</td>
<td>80</td>
</tr>
<tr>
<td>6.7</td>
<td>Extended Facebook dataset for User whose id is 1</td>
<td>81</td>
</tr>
<tr>
<td>6.8</td>
<td>Extended Facebook Dataset Statistics</td>
<td>82</td>
</tr>
<tr>
<td>6.9</td>
<td>Sample results - Common Movies Interest between the Friends of two Facebook Users identified by ids 305 and 389</td>
<td>82</td>
</tr>
<tr>
<td>6.10</td>
<td>More Comparison between the Similarity Measures for two Facebook Profiles identified by ids 441 and 134</td>
<td>83</td>
</tr>
<tr>
<td>6.11</td>
<td>Correlation between Different Measures</td>
<td>83</td>
</tr>
<tr>
<td>6.12</td>
<td>Comparison between Wordnet-Cosine and Random Predictors for DistT = 10</td>
<td>84</td>
</tr>
<tr>
<td>6.14</td>
<td>Comparison between Random Accuracy and Wordnet-Cosine distance Accuracy when Distance Threshold = 60 and for different values of % edges deleted</td>
<td>84</td>
</tr>
<tr>
<td>6.15</td>
<td>Difference between the modified Wordnet-Cosine distance and the Random Graph adjacency matrix</td>
<td>84</td>
</tr>
<tr>
<td>8.1</td>
<td>System Specification</td>
<td>88</td>
</tr>
<tr>
<td>8.2</td>
<td>Used software</td>
<td>89</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 The Similarity Problem in Social Network

1.1.1 Social Network and background

Social networks, which emerged as a form of complex network, are used widely by different types of individuals. The first social network website, SixDegrees.com, was launched in 1997. It is with the appearance of Facebook, launched in 2005-2006, that interest in social network grew tremendously followed by growth of recommender systems, which in fact, correspond to a social network as well [21]. Myspace was launched in 2004 [1] before Facebook, and it has been used since then by many users especially new musicians.

Social Networks allow people and/or companies to manage their online presence. People connect and share their personal details, including likes and dislikes. Many social networking websites have been created (such as Facebook [2], Twitter [3] and Myspace [4]) and they vary in the services which they provide. However, they mainly allow users to share pictures and videos. Typically, users of social networks seek to connect with other users (e.g., friends in Facebook). However, given the large number of social network users to choose from, it is usually difficult for any one of these to find and select friends to connect with. Various tools for friend suggestions have been developed to assist in this selection, including methods that use semantic measures of similarity between user


1
profiles in the network. In addition to suggesting friends or followers in social network, methods for suggesting items to buy may be investigated in retail websites (such as Amazon[5]). Such methods are based on an implicit social network of users who buy similar items, as tracked by Amazon.

Facebook allows users to search for friends within a geographical area (e.g. Cincinnati), university, work place, living place, as well as by using other attributes. Some social networks websites ask for the login information of email services (Yahoo or GMail) in order to connect with the contacts from the user email account, and build the friends list for the new users (newbie).

Account

A social network user must have an account which is created by choosing a username, password, and providing the website with a valid email address. After that, the newly registered user fills up the profile with personal information and connects with friends. The privacy setting can be customized using various privacy options (e.g. in Facebook: Family, friends, close friends, etc).

Profile

The user profile page, lists, aside from his/her personal information, the connections (friends in Facebook, followers in Twitter, etc) that the user has. In particular, in Facebook, the profile shows, depending on the privacy setting chosen by the user, the personal information (birthday, name, and possibly the phone number), likes, friends, interests (movies, TV shows, etc), and the wall posts. Each user has a wall that shows friends' posts, in which the user was tagged, and also the profile owner’s posts. Twitter profile shows the tweets by the user, favorites, connections (followers and followed), pictures and videos posted by the user. A twitter user can protect his/her tweets from being visible to the public by adjusting the corresponding settings so that only confirmed followers can see them.

Friends

Facebook limits the number of friends to 5000[6] while Twitter has an initial limit of people to follow of 2000. If the user wants to follow more than this limit, (s)he can follow only 10% of his followers.

(for example a user with 15,000 followers is allowed to follow 3000).

Search

In order to search for friends in Facebook, the user can specify the value of different search criteria such as: Name, Hometown, Current City, High School, Mutual Friend, College, Employer, and Graduate School. A Facebook or Twitter user can also add a friend by providing his/her email address, as mentioned previously. The objective of the work described here is to improve the "People you May Know" service in Facebook, "Customers who bought this item also bought” service in Amazon, and "People who liked this also liked” service in IMDB[8] and any similar feature which helps in avoiding searching a large amount of data and tackles the problem of finding the right match between user profiles, by suggesting those who might be similar or items that might be of interest.

Messages, Notifications, and Special Features

Most Social Networking websites support messaging between users which allows them to have a private conversation in addition to the public communication they post on their profile pages. In Facebook and Twitter more than one person can be selected to receive a message by simply typing their usernames in the message-to field.

In Facebook, automatic notifications let users know about activities of their connections. These notifications can be customized to meet the user needs. For instance, one can choose whether to receive notification from a group (every time a user posts in that group) or an individual. Basically "Following” a user leads to receiving notification of the user actions. Facebook allows users to view the latest activities log: comments, friendship request and acceptance, likes, etc. Facebook provides the users the option for creating special purpose groups which can be closed or open. Members of the same group can share their post on the group and communicate with each other.

[8]www.imdb.com
Limitations and Fellowship vs Friendship

Twitter users fall into two categories, content providers (also known as followees) and content consumers (also known as followers). A Twitter user is usually both a follower and a followee. Moreover, in contrast to Facebook, Twitter limits the number of characters that can be posted: people post tweets limited to 140 characters, Facebook status messages have a much larger limit (as of 2011 they were extended to approximately 60,000 characters\(^9\)). In Facebook a link/friend connection between two users can be established only upon mutual agreement, while in Twitter, one can follow people regardless of their consent. Table 8.2 lists some differences between Facebook and Twitter.

<table>
<thead>
<tr>
<th>criteria</th>
<th>Facebook</th>
<th>Twitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>connection</td>
<td>friends</td>
<td>followers/followees</td>
</tr>
<tr>
<td>posts</td>
<td>status</td>
<td>tweet</td>
</tr>
<tr>
<td>support groups</td>
<td>(\checkmark)</td>
<td>(\times)</td>
</tr>
<tr>
<td>messages</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
</tr>
<tr>
<td>connection approval</td>
<td>(\checkmark)</td>
<td>(\times)</td>
</tr>
<tr>
<td>customizable wall</td>
<td>(\checkmark)</td>
<td>(\times)</td>
</tr>
</tbody>
</table>

1.1.2 Similarity

Similarity assessment is one of the most common operations underlying information processing in humans. This assessment is usually based on the components/attributes of the two objects. For example, such components may be letters in strings, words in documents, attributes or neighbors of nodes in a graph.

Applications of finding similarity range from recommender systems to link prediction and clustering. The semantic relationship between two words can be investigated using a conceptual taxonomy, which defines relationships between the different types of words. Such a taxonomy is used by Wordnet (more about Wordnet can be found in Section 4.2.1) which supports semantic analysis of linguistic phrases. In contrast to syntactic similarity, which is based on exact matching of words, and therefore similarity is 0 or 1, the semantic similarity is based on meaning of words, and

\(^9\)http://mashable.com/2011/11/30/facebook-status-63206-characters/
therefore results in similarity value in \([0,1]\), which helps in enhancing information retrieval.

This work investigates a semantic similarity measure on the content of social network. It defines a unified similarity measure where a node profile is represented as a *semantic vector of semantic entities* extracted according to Wordnet. A new similarity measure is proposed by evaluating the *cosine similarity* between these semantic vectors. Its performance is compared with other similarity measure on several datasets.

### 1.2 Organization of the Thesis

This thesis is organized as follows: first, an introduction about Social network and similarity was included in Chapter 1, which is followed by details about an initial experiment in Chapter 2. Chapter 3 surveys different types of similarities. Then a comparison between four selected similarity measures is included in Chapter 4 as well as some results. Chapter 5 includes Algorithmic description for each of the four similarity measures. After that, in depth overview of the two sets of experiments conducted to evaluate the different similarity measures is included in Chapter 6. Finally, the conclusion and future work.
Chapter 2

Initial Experiment

2.1 Initial Experiment

To begin with, an experiment was conducted to assess the proposed similarity measure, Wordnet-Cosine, against the Occurrence Frequency (OF) similarity (more about the OF can be found in Section 4.2.3). For this purpose, two data sets were used as follows: Facebook dataset, containing the movies interest of users, found in a list known as SkullSecurity [5], and a dataset from the DBLP site [2], which contains the publication titles for scientific/scholarly papers.

2.1.1 Related Work on Similarity

Analysis of similarity between Facebook profiles can be assessed from the study of keyword similarity [12]. To find the relationship between the keywords, they were arranged in a hierarchical structure to form trees of possibly different heights. In the forest model, more than one tree was generated for each profile. A set of heuristics to retrieve the related words was used. The four heuristics are:

- Base: The tree is composed only of the initial keyword.
- Holonyms/Meronyms (HM): The whole/part description
- Synonym/Similar (SS)
- ALL: using all of the previous descriptions

Wordnet was used to find the semantic relationship between the words. [12]
The semantic distance between profiles is very important to this process, as it has been shown that the similarity between profiles deteriorates as the distance between them increases. Manhattan and Euclidean distance are independent of the distribution underlying the data set. It was shown that when similarity is dependent on this distribution, no single measure is superior [14].

The approach taken in this thesis does not depend on the distribution underlying the dataset. Barbara, et al proposed a method of network similarity that only depends on the structure of the dataset and profile similarity measures, and a method for inferring missing items. They used the occurrence frequency because it produces values in the interval [0, 1] according to how similar profiles are, based on the frequency of the items in the dataset [8]. In contrast to the Inverse Occurrence Frequency (IOF), less frequent mismatches are assigned lower similarity when using the Occurrence Frequency (OF), while mismatches on high frequent values are assigned higher similarity. The similarity measures are classified according to which part of the similarity matrix they fill with weighted values (not equally assigning a single value of 0 or 1). The diagonal similarity measures assigns 0 as the similarity values for all mismatches and a weighted similarity for matches. The off diagonal measures, assign 1 as the similarity values for all matches and a different weights to mismatches. Both ways produce weighted values for matches (diagonal) and mismatches (offdiagonal). OF is one of the measures that has best performance in detecting outliers. Outliers detection could be used as a measure for evaluating the approach taken in this thesis, by adding the outliers, feature values that are far away from the top concept in the semantic hierarchy, to the test set. Then the k nearest neighbors are found using the Wordnet-Cosine (proposed in this thesis) algorithm and the occurrence frequency method, in a way similar to [14].

Link prediction can be approached using three measures based on, respectively, (1) the topological structure of the data, (2) the profile information, and (3) combination of both [42]. It has been found that algorithms based on topological structure of profiles and techniques based on user-created similarity perform better at suggesting new links/friends.

The Inter-Profile Similarity (IPS) algorithm [53] utilizes Natural Language Processing (NLP) to find similar social network profiles based on the approximate matching of phrases. It includes ProfileSimilarity, a measure which takes two short snippets from two profiles (respectively, A and B) and performs word sense disambiguation first. Then it finds the meaning of each snippet, which will be used to find the similarity between them, a value in [0, 1]. A user study was conducted to
evaluate the algorithm. The authors report that the approach suffers from “few shortcomings that need to be solved”. The IPS algorithm was also compared with the simple intersection approach of finding similar profiles. The main concern was the change in the semantic similarity with the distance between profiles. Enhancement of the profile semantic similarity was one of the motivations for the approach presented in this thesis.

Similarity evaluation is an important aspect for recommending online communities to the users. For example, in [54], six different measures of similarity (including L1- and L2- norms) for recommending online communities were evaluated, based on using the information of visit and join to communities. However, no semantic information or NLP tools were used.

An adaptation of the Occurrence Frequency is described in [9] which produces 1 for identical values and a non-zero value for distinct item values (dissimilarity). In addition, the approach takes into account the distribution of the values in the dataset.

**Missing Values**

The simple approach of giving high similarity value for profiles with similar feature values, and low similarity for dissimilar values, results in low similarity values for profiles with missing values. Using the network information to predict missing values is not accurate and it cannot distinguish between cases of missing values, and the “doesn’t apply” situation for an attribute. To calculate $f(x)$, the frequency of feature value $x$, instead of the number of records in friends list that have the value $x$ (as done in [9]), in this thesis, the number of records in the dataset with the value $x$ is used, which leads to higher OF similarity values.

**Clustering**

Clustering techniques can be used for link suggestion/prediction. However, similarity/distance evaluation underlies such techniques. In [16] graph clustering and relational clustering were used for link suggestion. To take into account profile information, dummy vertices were added to the original graph to represent attribute values, converting this way, profile information into graph structure information. A unified neighborhood random walk was then performed on the resulting graph. Node profile information was used in a syntactic manner (exact matching of words), and no NLP tools (such as a tagger) were used. Semantic similarity could be added to this approach
resulting in a graph with a simpler structure (e.g., smaller number of nodes).

In another departure from previous work, this thesis investigates semantic relationship between attribute entries in the social network, not only between keywords. Therefore the category of a word must be found. This can be accomplished by using a tagger, a program which tags a word by its part of speech category \[3\]. The categories used in this study are: NN (noun, proper, singular or mass), NNP (noun, proper, singular), NNS (noun, common, plural), and NNPS (noun, proper, plural) \[6\]. These part of speech tags are used to assess profile similarity. This research improves on finding the similarity between profiles using the semantic distance between attribute entries by using Wordnet as a lexical database. The approach taken here is illustrated on two datasets, Facebook and DBLP. Before proceeding further, Example 2.1.1 illustrates similarity-based link prediction in a network.

**Example 2.1.1** Consider the network represented by the graph \(G = (V,E)\) of Figure 2.1. For each node, its profile is represented by the string of characters shown. For this example, the similarity of two profiles, \(P_1, P_2\) is defined as the Jaccard similarity

\[
J(P_1,P_2) = \frac{|P_1 \cap P_2|}{|P_1 \cup P_2|}
\]

Table 2.1 shows the pairwise similarities computed according to this formula.

**Table 2.1**: Pairwise similarity values for the network in Figure 2.1

<table>
<thead>
<tr>
<th>Nodes</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1/4</td>
<td>0</td>
<td>1/6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1/4</td>
<td>0</td>
<td>0</td>
<td>1/5</td>
</tr>
<tr>
<td>4</td>
<td>1/4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1/4</td>
<td>0</td>
<td>1</td>
<td>1/4</td>
<td>0</td>
<td>1/3</td>
</tr>
<tr>
<td>6</td>
<td>1/6</td>
<td>0</td>
<td>0</td>
<td>1/3</td>
<td>1/4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1/5</td>
<td>0</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

*Figure 2.2* shows the network augmented with (labeled) links reflecting the pairwise similarities between the pairs (Node 5, Node 8), (Node 5, Node 6), and (Node 6, Node 1).
2.1.2 Datasets

The initial experiment was based on two datasets as follows:

Facebook Dataset

Facebook is a well known social networking website. At the end of 2014 Facebook reached 1.4billion regular users per month. The Facebook dataset considered in the experiments contains 585 profile pages from Facebook (row data before the introduction of the Facebook timeline), downloaded data (9/2011 - 12/2011) from Skull Security, which has a list of publicly available Facebook URLs. More specifically, Dataset.txt (Facebook Dataset) contains all the movies interest for different Facebook profile numbers. The format of the dataset is as follows:

- "Profile id"
- "Movies of interest entered by the user identified by the profile id" separated by comma.

---

Table 2.2 shows the statistics of this data set and Figure 2.3 shows the frequency of the top 20 movies in it.

**Table 2.2: Statistics of the Facebook Dataset**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Facebook profiles</td>
<td>585</td>
</tr>
<tr>
<td>Average movies entries per profile</td>
<td>2.0</td>
</tr>
<tr>
<td>Number of movies entries for all profiles</td>
<td>1744</td>
</tr>
<tr>
<td>Maximum movies entries</td>
<td>8</td>
</tr>
<tr>
<td>Most Common Genre type</td>
<td>Which is the genre type unknown</td>
</tr>
<tr>
<td>Minimum movies entries</td>
<td>1</td>
</tr>
<tr>
<td>Different movies count</td>
<td>1103</td>
</tr>
</tbody>
</table>

*a* This was calculated as: (summation of the number of movies entries for different Facebook profiles) / number of Facebook profiles.

*b* Maximum summation of genre category for all profiles

*c* This was calculated as the number of rows in the dataset

The authorship dataset (DBLP)

DBLP website provides a collection of computer science papers. It can be accessed using the URL [http://dblp.uni-trier.de/](http://dblp.uni-trier.de/). The dataset used contains 3566681 nodes and we only used 585 [2].

### 2.1.3 Initial System Architecture

The system consists of three main components, the tagger, an encoder, and profile matcher. See Figure 2.4
The tagger finds the part of speech tag for each word in the sentence after removing the stop words. The encoder, which communicates with Wordnet to find the hypernym that belongs to a particular synset, finds the distance from each selected word to the top entity and encode this distance in the representational vector. Then the cosine similarity is computed using equation 2 below.

2.1.4 Results of Initial Experiments

The OF algorithm [8], and the Wordnet-Cosine approach were implemented using Java. Figures 2.5 and 2.6 show the results of these two approaches for the Facebook and DBLP data sets respectively. For both dataset, using the OF, most of the data are similar: 501 out of 583 pairs, of the Facebook data set, are evaluated to have similarity equal to 1, the remaining pairs being distributed in two similarity bins. For the same data set, the Wordnet-Cosine approach produces a wide range of similarity values (see Figure 2.5). The same results can be observed for the DBLP data set: all of the 584 pairs considered are output by OF to have similarity equal to 1. Again, for the same pairs, Wordnet-Cosine produces a wide range of similarity values (see Figure 2.6). Moreover, it can be observed that similarity values have a Normal-like distribution with mean around 0.6.

2.1.5 Conclusion for Chapter 2

The initial experiments with the Wordnet-Cosine and OF similarity measures show that the former is more sensitive, resulting in a wide range of values for the two data sets considered. By contrast, the OF measure, which is known to achieve good performance in detecting outliers, is of very
limited use when the data set is more homogeneous. In other words, OF seems to detect very well, cases of very low similarity, for the rest assigning almost always similarity close to 1. Often, exact similarity values may not be of interest. However, having a wider range of such values makes it possible to rank them, and therefore, from this point of view Wordnet-Cosine is more desirable than OF.
Chapter 3

Similarity Between Nodes in a Social Network
3.1 Similarity in Social Networks

In this chapter, several similarity measures are surveyed. Finding similar nodes (node similarity) in a social graph is the solution for many social graph problems such as link prediction, and community formation.

Similarity between nodes can be traced back to similarity between strings. Similarity is often defined as a decreasing function of a distance measure. *editDistance* [36] and *trigrams*, shown in equations (3.1) and (3.2) respectively, are among the most used distance measures [11].

\[d_{edit}(x, y) = \min\{\gamma(S) | S \text{ is an edit sequence taking } x \text{ to } y\}\] (3.1)

\[d_{tri}(x, y) = \frac{|\text{tri}(x) \cap \text{tri}(y)|}{|\text{tri}(x) \cup \text{tri}(y)|}\] (3.2)

where \(\text{tri}(x)\) refers to the collection of trigrams (ordered substrings of length 3) of \(x\), and \(|\text{tri}(x)|\) denotes the number of trigrams of \(x\). Then the similarity measure corresponding to the distance measures shown in (3.1) and (3.2) is defined as in equations (3.3) respectively [36] [14].

\[sim_a(x, y) = \frac{1}{1 + d_a(x, y)}\] (3.3)

where \(a \in \{edit, tri\}\).

Finding similar profiles to a network node, has been studied by many researchers [59], [26], [44], [55]. Having systems or services that can automate this task helps in avoiding the need to search in a large network of data. Moreover, it has many applications in social networks (e.g., Facebook, Linkedin) as well as other networks (e.g., recommendation systems). Many social networks websites such as Facebook, MySpace, Twitter, YouTube, and Orkut [1] are very popular. For instance, by the end of 2010, Facebook had in excess of 1.2 billion users [22]. People have turned to such websites to exchange posts and messages and social network users can express their approval of a post by liking (in Facebook) or favoring that post (in Twitter). Such huge amount of information sharing raises questions concerning the privacy of the individual users [19]. Psychology plays an important role in driving people to take part in the social networks, given their definition, characteristics, and

---

1. [www.facebook.com](http://www.facebook.com), [www.myspace.com](http://www.myspace.com), [www.twitter.com](http://www.twitter.com), [www.youtube.com](http://www.youtube.com), [www.orkut.com](http://www.orkut.com)
motivation to join such networks (see for example [25] and references therein).

Formally, social networks are represented as a graph of nodes and edges. Nodes represent profiles and edges represent connections between two nodes. Similarity between nodes could be based on either nodes attribute (textual) and/or edges/links (structure). Similarity measures in a graph vary; some similarity measures [29] are based on the commonality between the nodes in the graph (use the neighbors of nodes in the graph), while other similarity measures are based on link similarity; these vary in the length of the path they consider [31]. For example, the link similarity described in [31] considers paths of length greater than 2 path of length exactly 2. Others define the link similarity based on the number of paths of varying length between such nodes [45].

Lexical database or ontologies, such as Wordnet and SNOMED CT [39] and [50], may be used in finding the similarity between items expressed as free text or keywords. Wordnet is considered more general than the SNOMED CT ontology which is mainly in the health care domain. Moreover, the work described in [50] finds the similarity between sentences not just words, so tools of natural language processing (NLP), such as a tagger, are included.

Several similarity measures have been introduced including, Jaccard (biology) [28], cosine, min [31], Sorensen, Adamic Adar [7], and resource allocation [61]; PageSim, a method to measure the similarity between web documents was proposed in [37], based on PageRank score propagation. PageSim was evaluated against standard information retrieval similarities TF/IDF, which were considered to be the ground truth. Most of the similarity measures described in the literature are knowledge dependent. However, the authors in [36] describe a knowledge independent definition of similarity in terms of information theory. A list of similarity properties (axioms) was included in [15].

3.2 Semantic Similarity

Using Wordnet or more special lexical database in finding similarity has been researched in many papers [34], [27]. Different types of semantic similarity exist including

1. feature based

2. information content (which is based on the frequency of words in the dataset)

3. a combination of both (hybrid)

4. ontological measures (which take into consideration the number of nodes/edges between two concepts) [20].

The feature based semantic similarity uses the glossary for each concept (term) given by Wordnet. Each concept in Wordnet, is defined and thus this definition can be considered to find the similarity in what is known as feature similarity. The path similarity uses the structural relationship between the concept (taxonomy, or ontological hierarchy). Edge counting measures suffer from irregularities in path lengths so extra attention must be taken when using them.

The information content similarity relies on the distribution of the concepts in the dataset. In particular, it combines statistics and taxonomy structure [30]. The performance of Information Content similarity measures is better than the measures which are only edge based as it has been reported in [30]. Moreover, research that uses Wordnet mainly only considers the is-a relationship (hyponymy/hypernymy) [32].

The reader can refer to [49] for further comparisons between the three similarity measures. The authors have indicated that measures which are based on corpora statistics (information content) require intensive computations, and therefore are impractical when the corpora is very large.

Wordnet is a free lexical database that organizes English words into concepts and relations between them. English words, whether they are nouns, verbs, adjectives, or adverbs, are represented as a hierarchy of synsets that are connected by a relationship between them. On the other hand, each concept in Wordnet can have more than one synset, which is determined by the hypernym (is-a) hierarchy.

Wordnet-Similarity, is a Perl package that uses Wordnet to find the similarity between concepts [47]. The developers of the package included the implementation of three information content similarity measures and three edge based similarity measures.

Two research studies, that use ontology in conjunction with cosine similarity, are detailed in [39] and [50]. However, the main difference between the two is basically in the type of ontology that has been used ( [39] uses Wordnet while [50] uses SNOMED CT). The second difference comes from the type of operands for the similarity measure: free text or merely keywords. The work described in [50] uses tools of Natural Language Processing, more specifically it uses the Stanford Tagger [3].
3.3 Problem description and evaluation metrics

The problem of similarity can be precisely described as follows. Given a graph representation of a social network, with two kinds of information - node attributes and link structure - find all pairs of similar nodes based either on node profiles (node attribute) or link (structural attributes).

More formally, given a graph $G = (V, E)$, where $V$, the set of vertices which represents nodes, and $E$ the set of edges which represents the links in the network, find all pair of similar vertices $(v_i, v_j), v_i, v_j \in V$ using the features of $v_i$ and $v_j$ or the links $e_{i1}, \ldots, e_{in}$ and $e_{j1}, \ldots, e_{jm}$ for both the vertices $v_i$ and $v_j$.

Each node $v_i$ in the graph can be described using a set of values for each attribute $f_i$ such that $f_i \in F$, where $F$ is the set of all attributes of each node. Alternatively, such nodes can be also described as having a link $e_i$ to a particular node in the graph.

The output of the similarity system is a value which represents the similarity between each pair of nodes in the graph. The exact values of these similarities differ depending on the type of similarity measure being used. In general, of interest is the ranking of similarity evaluations, not their exact values.

3.4 Motivation for finding similarity

Finding the similarity between objects can be used to solve many kinds of problems: items or friends recommendation based on the commonality [60], problems in clustering, collaborative filtering, and search engines [23]. Different similarity measures have been used in biology, ethnology, taxonomy, image retrieval, geology and chemistry [17], as well as in the biomedical field [39]. Neighborhood search, centrality analysis, link prediction, graph clustering, multimedia captioning, related pages suggestion in search engines, identifying web communities, friends suggestion in friendship network (Facebook or MySpace), movies suggestion, item recommendation in retail service, scientific and web domains in general, are all different kinds of applications for similarity measures in data. An example of using similarity for clustering is collaborative filtering [23]. Zhou, Cheng and Yu proposed an algorithm to clustering objects using attributes and structure where the attribute of a node and the structure are independent [63]. Moreover, finding similarity between objects was also investigated in information retrieval [37].
3.5 Node and other similarity measurements

Similarity in network can be based on different elements depending on the type of results that are being sought and the type of information available. For instance, Node and edge similarities come into presence when the graph structure is considered. On the other hand, general ontologies or domain knowledge can be used when semantic similarities are considered. Furthermore, similarity between words, documents, or between profiles (nodes) are used [55], [43]. Similarity measures used in clustering [63] vary according to whether they are content-based, title-based, or keyword-based [58].

Structure similarity (link-based). As the name implies, the structure similarity examines the graph (links). These links may represent friendship, co-authorship, payment, etc. It has been reported that, when compared with the human judgment, structure similarity produces better results [33]. Such a similarity measure, based on node neighbors is described in [31].

Content similarity (text-based). As opposed to structural similarity, content similarity consider the node data in finding the similarity. There are so many kinds of attributes that can be associated with nodes such as: birth date, college, hobbies, movies interest, and age. Crowd-sourcing can be used to collect vast number of tags that represent content such as movies which can also be considered as a type of user-defined tags. These tags can be used to build a new recommender system [48].

Keyword similarity (word-based). A selected subset of the possible words, known as keywords that can occur in the node attributes may be used to find nodes similarity. An example of a keyword similarity is the forest model described in [13], in which the keywords are organized into a forest model, on which Wordnet is used subsequently to find the similarity between keywords.

Table 3.1 shows a snapshot of the Facebook dataset (see Section 2.1.2 for more details about the dataset) combined with synthetic Friends id’s data which was randomly made just for the sake of explanation of the node and edge similarities. Each profile is represented as a set of movies interest and a set of Facebook friends id’s. Profile similarity between these two nodes must be made in terms of the list of movies in each of these profiles.
Table 3.1: Two Facebook profiles

<table>
<thead>
<tr>
<th>Profile ID</th>
<th>Profile</th>
<th>Friends IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxxxx773.html</td>
<td>Comedy, Action films American, EL EL</td>
<td>1, 2, 3, 10</td>
</tr>
<tr>
<td>xxxxxxx432.html</td>
<td>Haunted 3D, Saw, Transformers,</td>
<td>3, 4, 5, 9, 10</td>
</tr>
<tr>
<td></td>
<td>Pirates of the Caribbean, Mind Hunter</td>
<td></td>
</tr>
</tbody>
</table>

3.5.1 Node Similarity

Four similarity measures, Wordnet-Cosine, WordFrequencyVector(WFV), SemanticCategories, and SetSimilarity, were compared in [50]. For the first three, the node profile is represented as a vector, and each similarity measure encodes the profile in a different way.

For the Wordnet-Cosine similarity, a profile $X$, is represented by the vector $D_X = [D_{x1}, \ldots, D_{xn}]$, where $D_{xi}$ denotes the distance between the ith word in the user profile to the highest (concept) in the hierarchy of concepts obtained using Wordnet. The Wordnet-Cosine similarity of two profiles, $X$ and $Y$, is then defined as the cosine of the associated vectors as shown in equation (3.4).

$$Sim_W(X,Y) = \cos(D_X, D_Y),$$ (3.4)

For the WFV similarity measure, a node profile $X$ is represented by the vector $F_X = [F_{x1}, \ldots, F_{xn}]$, where $F_{xi}$ denotes the frequency of the ith word in the dataset. The WFV similarity of two profiles $X$ and $Y$ is then defined as the cosine of associated vectors as shown in equation (3.5).

$$Sim_{WFV}(X,Y) = \cos(F_X, F_Y),$$ (3.5)

The SemanticCategory similarity measure is defined as the cosine of frequencies of semantic categories, as shown in equation (3.6).

$$Sim_{SC}(X,Y) = \cos(SC_X, SC_Y),$$ (3.6)

where

$$SC_X = [f_A(X)|A \in \{NN, NNS, NNP, NNPS\}]$$

and $f_A(X)$ denotes the frequency of $A$ in $X$. 

20
Finally, the SetSimilarity is defined on the basis of the sets of parents of the words in the profiles, as shown in equation (3.7).

\[
Sim_S(X, Y) = \frac{|S_X \cap S_Y|}{|S_X \cup S_Y|},
\]

(3.7)

where \( S_X = \{S_{xi}|i = 1, \ldots, n\} \) is the set of parents for the \( i \)th word in the user profile \( X \) obtained by using Wordnet.

### 3.5.2 Edge similarity

Equations (3.8) - (3.11) shows some of edge similarities, where \( \Gamma(X) \) denotes the set of neighbors of \( X \), and \( K_X \) is the degree of node \( X \):

\[
Sim_{Salton}(X, Y) = \frac{|\Gamma(X) \cap \Gamma(Y)|}{\sqrt{K_X \times K_Y}}
\]

(3.8)

\[
Sim_{Jaccard}(X, Y) = \frac{|\Gamma(X) \cap \Gamma(Y)|}{|\Gamma(X) \cup \Gamma(Y)|}
\]

(3.9)

\[
Sim_{HPI}(X, Y) = \frac{|\Gamma(X) \cap \Gamma(Y)|}{\min\{K_X, K_Y\}}
\]

(3.10)

\[
Sim_{HDI}(X, Y) = \frac{|\Gamma(X) \cap \Gamma(Y)|}{\max\{K_X, K_Y\}}
\]

(3.11)

Table 3.2 shows the node similarities (Wordnet-Cosine, SetSimilarity, SemanticSimilarity, and Word FrequencyVector) and edge similarities (Salton, Jaccard, High Promoted Index, and Hub Depressed Index) for the two Facebook profiles shown in Table 3.1. As it can be noted from Table 3.2, with the exception of the SetSimilarity, in general, node similarities have higher similarity values than edge similarities. The highest node-similarity is Wordnet-Cosine and SemanticSimilarity, followed by WFV similarity and SetSimilarity. This is due to the fact that Wordnet-Cosine captures the semantics/meaning of terms/words and hence it has the capability to recognize similarity on a different level than just the exact match of words. The JaccardSimilarity and SetSimilarity are both based on sets (intersection and union). However, the former uses sets of neighbors (friends in Facebook) while the latter uses sets of semantic features of profiles.
Table 3.2: Node and Link similarities

<table>
<thead>
<tr>
<th>Node Similarity</th>
<th>Wordnet Cosine</th>
<th>Set</th>
<th>Semantic</th>
<th>Word Frequency Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.862795963</td>
<td>0.0659340066</td>
<td>0.877526909</td>
<td>0.74900588</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Link Similarity</th>
<th>Slaton</th>
<th>Jaccard</th>
<th>Hub Promoted Index</th>
<th>Hub Depressed Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slaton</td>
<td>0.423</td>
<td>0.285</td>
<td>0.5</td>
<td>0.4</td>
</tr>
</tbody>
</table>

With a threshold $\alpha = 0.5$ applied to the similarity values, then only SetSimilarity does not return the two profiles as similar; all other similarity measures return the two profiles as similar.

3.5.3 Global Structural Similarities

Structural similarities can be classified into [38]:

- local vs. global
- parameter-free vs. parameter-dependent
- node-dependent vs. path-dependent

In the remainder of this section, several structural similarities are discussed, as follows:

1. SimRank
2. SimFusion
3. P-Rank
4. E-Rank
5. Vector Space (cosine similarity, and pearson correlation coefficient).
6. GroupRem
7. PageRank and PageSim

*SimRank* considers two objects to be similar if they are related to similar objects [29]. *SimFusion* [57] considers evidence from multiple sources when trying to find the similarity between two objects.
One of the differences between SimRank and SimFusion is that SimFusion uses two a random walker approach \cite{57}. SimRank considers two entities to be similar if they are both referenced by similar entities. \(P - \text{Rank} \) \cite{62} considers both in-links and out-links in contrary to SimRank. On the other hand, \(E - \text{rank} \) considers entities to be similar if also they reference similar entities. SetSimilarity fails to recognize similarity between objects which are represented in a hierarchical manner, and it results in 0 similarity value between objects of different heights even though they may be similar.

Two of the vector based similarity measures are cosine similarity and pearson correlation coefficient. A comparison between six different similarity measures, including cosine index and pearson correlation coefficient, is detailed in \cite{61}.

GroupRem is a group-based similarity measure computed on movies tags and popularity \cite{48}. Several binary similarities between binary vectors were described in \cite{17}.

PageSim finds the similarity of web pages in search engine or web document classification. It was inspired by PageRank, and it was evaluated against Cosine TF/IDF \cite{37}.

The ”People you may know”, friends recommender in Facebook, is based on the friends of friends (path of length two). Several friend recommender systems have been described and compared using precision and recall \cite{55}, \cite{45}.

3.5.4 Conclusion for Chapter 3

Several similarity measures have been surveyed in this chapter. Table 3.3 and 3.4 compare a list of similarity measures, with respect to time and space complexity (Table 3.3) and with respect to whom they compared their work with, dataset, and performance (3.4). Additional comparisons can be found in \cite{36} and \cite{17}.
Table 3.3: Comparison between similarity measures

<table>
<thead>
<tr>
<th>Similarity Measure</th>
<th>Time</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimRank</td>
<td>$O(Kn^2d^2)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Improved SimRank</td>
<td>$O(k^4n^2)$ ($k \leq n$)</td>
<td>$k^2 \times n^4$</td>
</tr>
<tr>
<td>PageSim</td>
<td>$O(C^2)$, $C = kr$</td>
<td>$O(Cn)$, $C = kr$</td>
</tr>
<tr>
<td>E-Rank</td>
<td>$O(n^3)$, but more extensive evaluation to be considered in future work</td>
<td>future work</td>
</tr>
<tr>
<td>SimFusion</td>
<td>$O(Kn^2d)$, where $d$ is the number of iterations</td>
<td>$O(n^2)$, where $n$ is the total number of objects</td>
</tr>
<tr>
<td>P-Rank</td>
<td>$O(Kn^2d^2)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>FriendTNS</td>
<td>0.012sec, for $N=1000$, $k=10$</td>
<td></td>
</tr>
</tbody>
</table>
Table 3.4: Comparison between similarity measures

<table>
<thead>
<tr>
<th>Similarity Measure</th>
<th>Compared with</th>
<th>Dataset used</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimRank</td>
<td>Co-citation [52]</td>
<td>Research Index(^1)</td>
</tr>
<tr>
<td>Improved SimRank</td>
<td>SimRank</td>
<td>DBLP(^2); Image data (querying Google Image Search); Wikipedia(^3)</td>
</tr>
<tr>
<td>PageSim</td>
<td>SimRank; Cosine TFIDF as a ground truth</td>
<td>crawled Webpages(^4)</td>
</tr>
<tr>
<td>E-Rank</td>
<td>Enriches P-Rank by considering both in- and out-links</td>
<td>Enron Email dataset(^5), Citation Network(^6), DBLP(^7)</td>
</tr>
<tr>
<td>SimFusion</td>
<td>SimRank (detailed description) and tf × idf</td>
<td>Search click through log</td>
</tr>
<tr>
<td>P-Rank</td>
<td>Extends SimRank</td>
<td>Synthetic(^2)</td>
</tr>
<tr>
<td>Vertex similarity(^3)</td>
<td>Cosine similarity and SimRank</td>
<td>AddHealth data: study as part of the National Longitudinal Study of Adolescent Health</td>
</tr>
<tr>
<td>FriendTNS</td>
<td>RWR, Shortest Path, Adamic/Adar, FOAF</td>
<td>Facebook, Hi5, Epinion</td>
</tr>
</tbody>
</table>

\(^1\)http://www.researchindex.com Transcript of 1050 students at Stanford University;  
\(^2\)http://kdl.cs.umass.edu/data/dblp/dblp-info.html;  
\(^3\)http://www.wikipedia.org/  
\(^4\)http://www.cse.cuhk.edu.hk  
\(^5\)http://www.cs.cmu.edu/enron/  
\(^6\)http://snap.stanford.edu/data/  
\(^7\)http://www.informatik.uni-trier.de/ley/db/
Chapter 4

Semantic Similarity Measures
4.1 Similarity measures and Semantic

As it has been mentioned earlier, several similarity measures have been developed and they can be used in various applications. lexical database, such as Wordnet, can be used to assist in obtaining the semantic underlying the content of the node profile in networks. Such semantic content, can be used in calculating the similarity between the data. In this chapter, more about the semantic analysis (using Wordnet), and how it can be integrated into the proposed similarity measure, Wordnet-Cosine, is described.

4.2 Finding Similar Profiles

4.2.1 Wordnet

As already mentioned the current approach makes use of Wordnet, a free lexical database that organizes English words into concepts and relations, well-known for assessing semantic similarity. This section discusses in more detail the elements of Wordnet. English nouns, verbs, adjectives, and adverbs form hierarchies of synset where relations exist that connect them. Of the six relations defined in Wordnet, Synonymy, Antonymy, Hypernymy, Meronymy, Troponymy, Entailment, this study uses only Hypernymy.

Hypernym of a word

Informally, Hypernym of a word is its super class concept. It is equivalent to the is-a or kind-of relationships used in ontologies. The opposite of Hypernym is Hyponym which is the sub-class. Consider for example, the two senses of word ”comedy”:

- comedy as a ”humorous drama”
- comedy as ”comic incident”

For the first sense, comedy is a kind of drama, which is a kind of literary work. Therefore, literary work is a hypernym of drama, and drama is a hypernym of comedy [41]. The hierarchy determined by the hypernym relationship is called a synset. Therefore, based on the above, the synset for
comedy (with respect to the first meaning) is

\[
\text{Synset 1: } \text{[entity]} \leftarrow \text{[abstract entity]} \leftarrow \text{[abstraction]} \leftarrow \text{[communication]}
\]
\[
\leftarrow \text{[expressive style,style]} \leftarrow \text{[writing style, literary genre, genre]}
\]
\[
\leftarrow \text{[drama]} \leftarrow \text{[comedy]}
\]
\[
\quad \text{light and humorous drama with a happy ending}
\] (4.1)

while the Synset with respect to the second meaning is:

\[
\text{Synset 2: } \text{[entity]} \leftarrow \text{[abstract entity]} \leftarrow \text{[abstraction]}
\]
\[
\leftarrow \text{[communication]} \leftarrow \text{[message, content, subject matter, substance]}
\]
\[
\leftarrow \text{[wit, humor, humor, witticism, wittiness]} \leftarrow \text{[fun, play, sport]}
\]
\[
\leftarrow \text{[drollery, clowning, comedy, funniness]}
\]
\[
\quad \text{a comic incident or series of incidents}
\] (4.2)

**Wordnet Java API**

Wordnet offers a Java API that can be used to query the Wordnet data. The Java API was created by Brett Spell [4]. It uses the WordNetDatabase and retrieves the synsets of a word from it.

### 4.2.2 Semantic Tagger

A semantic tagger is a program that takes as input a natural language (English) sentence, and outputs the tag corresponding to the (syntactic) role of that word in the sentence. [3]. Tags denote semantic categories. The tags used in this study are shown in Table 4.1.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>noun, proper, singular or mass</td>
</tr>
<tr>
<td>NNP</td>
<td>noun, proper, singular</td>
</tr>
<tr>
<td>NNS</td>
<td>noun, common, plural</td>
</tr>
<tr>
<td>NNPS</td>
<td>noun, proper, plural</td>
</tr>
</tbody>
</table>

Table 4.1: Word tags and their descriptions [6].
4.2.3 The occurrence frequency similarity (OF)

The occurrence frequency similarity \[8 \] for two profiles \(D\) and \(D'\) is given by equation (4.3)

\[
OF(i_D, i_{D'}) = \begin{cases} 
1 & \text{if } i_D.n = i_{D'}.n \\
\frac{1}{B} \sum_{k=1}^{B} (1 + A \times B)^{-1} & \text{if } i_D.n \neq i_{D'}.n
\end{cases} \tag{4.3}
\]

where \(i_D\) denotes the value of attribute \(i\) in the profile \(D\), \(i_D.n\) denotes the value of the \(n\)th subfield for \(i_D\), \(N\) is the total number of item values, and \(f(\cdot)\) is the number of records; \(A = \log\left(\frac{N}{1+f(i_u.n)}\right)\), and \(B = \log\left(\frac{N}{f(i_x.k)}\right)\).

4.3 A unified Similarity Measure: Wordnet-cosine similarity

The Wordnet-cosine similarity between two profiles is defined in terms of the synsets obtained from these profiles according to the following steps:

1. Extract the text in the feature field (movies, title) if the data-set is not formatted well.

2. Apply Natural Language Processing - parse the text extracted to obtain its structure

3. Get the first synset for each extracted word using Wordnet.

4. Encode the word as follows:
   (a) Get all hypernym of the synset of the word (the first synset is used).
   (b) Find the distance from the word to the root of the synset.

5. Each feature field of a profile is encoded as a vector of such distances.

6. Apply cosine similarity between vectors of such distances.

Only the words \(w\) with tags \(t_w \in Tags\) of Table 4.1 are used in encoding the profile as a vector of distances. These distances separate each word \(w\) and the top hypernym (entity concept) in the conceptual hierarchical representation of \(w\). Each profile is represented as a set of word-tag pairs \((w, t_w)\), and Wordnet is used to retrieve the set of hypernyms of each word \(w\) thereby encoded as a vector of distances. More specifically, for each word \(w_i\), the distance between it and the top
hypernym is found and placed as the $ith$ component value of the vector representing the profile.
The distance $d_i = d(w_i)$ for each given word is computed according to equation (4.4)

\[
d(w) = \begin{cases} 
\text{dist}(w, \text{[entity]}) & \text{if } w \text{ is in Wordnet} \\
0 & \text{otherwise}
\end{cases} 
\quad (4.4)
\]

For example, given the word “comedy”, whose tag, $NN$, belongs to $Tags$ of Table 4.1, its encoding will be the distance between it and the entity concept, and this distance is equal to 7. For words that don’t have any hyponym, their encoding is 0. Words that have a tag $\notin Tags$ of Table 4.1 are ignored.

The encoding of a profile $D$ is a mapping $e : D \mapsto \mathbb{R}^k_+$ such that

\[
e(D) = (d_1, \ldots, d_k)
\]

The Wordnet-Cosine similarity between $D$ and $D'$, where $D$ and $D'$ are two profiles with encodings $e(D) = (d_1, \ldots, d_k)$ and $e(D') = (d'_1, \ldots, d'_k)$, is given by the cosine similarity between the encoding $e(D)$ and $e(D')$ of the two profiles, shown in equation (4.5).

\[
\text{Sim}(D, D') = \cos(e(D), e(D')) = \frac{e(D) \cdot e(D')}{\|e(D)\| \|e(D')\|}
\quad (4.5)
\]

The main motivation underlying this work is to convert the problem from processing unstructured data into finding similarity between real valued vectors. One issue remains unexplored that is which synset to use, as it has been mentioned earlier, the first synset was used. Therefore, more than one encoding of the profile can be obtained using different synsets.

### 4.4 Experimental Results

The approach described in the previous section is applied to a Facebook data set introduced in Section 2.1.2 as shown next.
4.4.1 Facebook profiles data-set

Table 4.3 illustrates the encoding of the Movie Attribute for three Facebook profiles. The content of the three profiles is as follows:

\[ D_1 = \{"00000000XXXXXX.html; Harry Potter, Transformers, Mr.& Mrs. Smith"\} \]

\[ D_2 = \{"100000002XXXXXX.html; Sherina’s Adventure"\} \]

\[ D_3 = \{"100000005XXXXXX.html; Love mein Gum, Maqsood Jutt Dog Fighter"\} \]

Encoding the three profiles using the Wordnet results in the following vectors:

\[ e(D_1) = (0, 7, 8, 8, 0); \quad e(D_2) = (0, 8); \quad e(D_3) = (5, 0, 7, 0, 6, 4) \]

Finally, the pairwise cosine similarities between the three profiles \( D_1, D_2, \) and \( D_3 \), based on their encoding vectors \( e(D_1), e(D_2), e(D_3) \) are shown in Table 4.2.

Table 4.2: The cosine similarity of vectors \( v \)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0000</td>
<td>0.4509</td>
<td>0.7126</td>
</tr>
<tr>
<td>2</td>
<td>0.4509</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>0.7126</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

As it can be seen from Table 4.2, the largest cosine similarity is between the 1st and 3rd profiles, followed by that between 1st and 2nd document. This corresponds to the first two smallest distances between the vectors \( e(D_1) \) and \( e(D_3) \), and \( e(D_1) \) and \( e(D_2) \).

4.4.2 Results

Java was used to implement all similarity measures including the OF similarity, described in [8], in two sets of experiments.

In the first set of experiments, the similarity was calculated between each adjacent nodes (consecutive rows) in the data-set using both the OF measure and Wordnet-Cosine approach. Table 4.4 illustrates similarity results for two profiles using both OF and Wordnet-Cosine.
Table 4.3: Illustration of Movie Attribute of Facebook profiles: their tags and Hypernyms.

<table>
<thead>
<tr>
<th>Profile 1: Movie Attribute</th>
<th>Harry Potter, Transformers, Mr. &amp; Mrs. Smith</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>Harry Potter, Transformers, Mr. &amp; Mrs. Smith</td>
</tr>
<tr>
<td>Tags</td>
<td>NNP, NNP, NNPS, NNP, CC, NNP, NNP</td>
</tr>
<tr>
<td>dist to root in synset</td>
<td>0, 7, 8, 8, ignored, 8, 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Profile 2: Movie Attribute</th>
<th>Sherina’s Adventure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>Sherina’s Adventure</td>
</tr>
<tr>
<td>Tags</td>
<td>NNP, POS, NNP</td>
</tr>
<tr>
<td>dist to root in synset</td>
<td>0, ignored, 8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Profile 3: Movie Attribute</th>
<th>Love mein Gum, Maqsood Jutt Dog Fighter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>Love mein Gum, Maqsood Jutt Dog Fighter</td>
</tr>
<tr>
<td>Tags</td>
<td>NNP, NNP, NNP, NNP, NNP, NNP, NNP, NNP</td>
</tr>
<tr>
<td>dist to root in synset</td>
<td>7, 0, 7, 0, 6, 4</td>
</tr>
</tbody>
</table>

Table 4.4: OF and Wordnet-Cosine similarity of two Facebook profiles along their Movie Attribute. Profile IDs are partially masked for privacy.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Facebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile-1 ID</td>
<td>10000006XXXXXX.html</td>
</tr>
<tr>
<td>Movies Interests</td>
<td>Captain Jack Sparrow, Meet The Spartans, Ice Age Movie, Spider-Man</td>
</tr>
<tr>
<td>Profile-2 ID</td>
<td>100000067XXXXXX.html</td>
</tr>
<tr>
<td>Movies Interests</td>
<td>Clash of the Titans, Ratatouille, Independence Day, Mr. Nice Guy, The Lord of the Rings Trilogy (Official Page)</td>
</tr>
<tr>
<td>OF Similarity</td>
<td>0.9472</td>
</tr>
<tr>
<td>Wordnet based similarity</td>
<td>0.1892</td>
</tr>
</tbody>
</table>

Table 4.5 summarizes the result of a survey done to find user similarity (not system) between the movies interest of profile 1 and profile 2 in Table 4.4. The participants were asked to rate the similarity between these profiles from -2 (not similar) to 2 (similar).

For profile 1 in the Facebook dataset, Captain Jack Sparrow is not a movie but it is a character in the Pirate of the Caribbean movie. Both Ice Age Movie and Spider-Man are adventure movies, and Meet the Spartans is a comedy and war movie. Spider-Man is also an action movie, and Ice Age is an animation movie. For profile 2, all movies of Clash of the Titans, Independence Day, and Mr. Nice Guy are action movies. Ratatouille is an animation movie. Mr. Nice Guy is a Chinese movie that stars Jackie Chan (See Table 4.6). The similarity between the movies in profile 1 and profile 2 using the OF is almost 1, while the similarity using the semantic Wordnet is equal to 0.1.

Figure 4.1 shows the result of applying the OF similarity and the Wordnet-cosine similarity.
Table 4.5: User rating of the similarity between profile 1 and profile 2 using Facebook dataset in table 4.4

<table>
<thead>
<tr>
<th>Person</th>
<th>Rate (-2 to 2 for similar)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person 1</td>
<td>1</td>
</tr>
<tr>
<td>Person 2</td>
<td>0</td>
</tr>
<tr>
<td>Person 3</td>
<td>2</td>
</tr>
<tr>
<td>Person 4</td>
<td>1</td>
</tr>
<tr>
<td>Person 5</td>
<td>2</td>
</tr>
<tr>
<td>Person 6</td>
<td>0.8 (70%)</td>
</tr>
<tr>
<td>Average</td>
<td>1.13</td>
</tr>
</tbody>
</table>

Table 4.6: Genres for movies listed in Table 4.4

<table>
<thead>
<tr>
<th>Movie</th>
<th>Genre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Captain Jack Sparrow (Pirates of the Caribbean)</td>
<td>Action, Adventure, Fantasy</td>
</tr>
<tr>
<td>Meet The Spartans</td>
<td>Comedy, war</td>
</tr>
<tr>
<td>Ice Age Movie</td>
<td>Animation, Adventure, Comedy</td>
</tr>
<tr>
<td>Spider-Man</td>
<td>action, adventure, fantasy</td>
</tr>
<tr>
<td>Clash of the Titans</td>
<td>action, adventure, documentary, family, fantasy, romance</td>
</tr>
<tr>
<td>Ratatouille</td>
<td>animation, comedy, family, fantasy</td>
</tr>
<tr>
<td>Independence Day</td>
<td>action, adventure, drama, romance, sci-fi, thriller</td>
</tr>
<tr>
<td>Mr. Nice Guy</td>
<td>action, comedy, crime</td>
</tr>
<tr>
<td>The Lord of the Rings Trilogy (Official Page),</td>
<td>adventure, animation, drama, fantasy</td>
</tr>
</tbody>
</table>

for all the node pairs connected by an edge in the data set. Using OF, most of the data are similar, with similarity value equal to 1. By contrast, using Wordnet-cosine, the similarity values are distributed over all the data having a peak value at 0.2.

Once more, the Wordnet-cosine was compared with the other similarity measures, which this time include set similarity (Jaccard index), semantic similarity and vector cosine similarity using the same data. Table ?? shows the difference in the similarity results for two profiles, using four similarity measures. The results for all four similarity measures on the Facebook data set, are shown in Figure 4.2.
Figure 4.1: Histogram of OF and Wordnet similarity using the Facebook dataset.

<table>
<thead>
<tr>
<th>Profile ID</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>132XXXXXXX.html</td>
<td>Comedy, Action films, American, El El</td>
</tr>
<tr>
<td>774XXXXXXX.html</td>
<td>Haunted 3D, Saw, Transformers, Pirates of the Caribbean, Mind Hunter</td>
</tr>
</tbody>
</table>

SIMILARITY MEASURES

<table>
<thead>
<tr>
<th>Similairty Measure</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wordnet cosine</td>
<td>0.862795963</td>
</tr>
<tr>
<td>Set similarity</td>
<td>0.0659340066</td>
</tr>
<tr>
<td>Semantic similarity (Syntactical)</td>
<td>0.877526909</td>
</tr>
<tr>
<td>Word frequency vector similarity</td>
<td>0.74900588</td>
</tr>
</tbody>
</table>

Figure 4.2: Histogram of the four similarity measures using the Facebook dataset.
4.5 Conclusion for Chapter 4

This chapter considered more in depth five different similarity measures - their definitions and performance were studied. In particular, the capability of the Wordnet-cosine similarity measure was highlighted. Recapping the procedure to evaluate the Wordnet-cosine similarity, the following steps were taken: profile data was processed to extract only the nouns; the profile data was then encoded into a vector using the distance between the extracted words and the top hypernym concept (entity). When compared with human-based similarity assessment (based on a small experiment) a strong agreement was found with the Wordnet-Cosine similarity.
Chapter 5

Algorithms and Implementations for Finding Similarity

5.1 Java algorithms for finding similarities

This chapter includes a description and a discussion of the similarity algorithms used. The algorithms were applied on Facebook, MovieLens, or DBLP datasets. All of the similarity measures were implemented using Java, this choice was influenced by the fact that Wordnet-cosine similarity, the proposed similarity, uses Wordnet API (Application Programmer Interface) [4] which is written in Java.

The code for finding each similarity measure was divided into three different types of methods. The dependency between such methods is illustrated in figure 5.1 and 5.2. As it can be seen from these figure, the GetResults_(similarity measure) method calls find_(similarity measure), which in turn calls getDistance_(similarity measure). The role of each method is as follows:

- GetResults_ iterates through all pairs of data (lines) in the dataset and invokes find_. It writes the similarity value between each pair of data into a text file.

- find_ invokes getDistance_ with two lines passed as arguments, and then it finds the similarity between the its arguments, and returns this similarity to GetResults_(similarity measure).

- getDistance_ encodes each line in the dataset as a vector according to the specification of each similarity measure.
Each similarity measure encodes each line in the dataset in a distinctive manner. However, all of the similarity measures encode each line as a vector with varying characteristics, some of which are shared among the different similarity measures. The detailed descriptions of how each line is encoded into a vector, based on the different similarity measures are as follows:

**Set Similarity.** Each line is encoded as a set-valued vector: each component is the set ancestors (parents) from a word in the profile, to the top entity in the conceptual hierarchy retrieved by Wordnet.

**Wordnet-Cosine similarity.** The encoding is similar, to some extent, to the encoding of the set similarity detailed in the previous item. However, each vector, which represents a profile, is created by finding the length of the path that connects every word in the line to the entity concept in the conceptual hierarchy retrieved by Wordnet.

**Semantic Categories.** Encodes each line as a vector that contains the frequency of words that belong to the categories NN, NNS, NNP, NNPS (so for instance, $v = 2$ NN, 3 NNS, 0 NNP, 0 NNPS).

**Word Frequency Vector.** Encoding is similar to the one made by the Semantic Categories where each component of the vector is the frequency of the word in the profile (not the frequency of its semantic category).

For an algorithm description of the encoding for each similarity measure, refer to the corresponding algorithms 4, 7, 10, and 13. Each similarity measure uses different criteria for calculating the similarity, and this can be summarized as follows:

**Set similarity** calculates the similarity between two profiles as the ratio of the size of the intersection of the sets representing each and the size of the union of such sets. Note: Recall that each set contains the parent set.

**Wordnet-Cosine, Semantic Categories, and Word Frequency Vector** similarities are all based on the cosine similarity of the corresponding vector representation.
The algorithm, shown in figure 1, is used to populate one variable, named DBlines, with the content of the dataset such that each of its components holds an element of the dataset that represent an entity (person, movie, etc). This may facilitate processing the dataset and achieve more organization within the algorithms, not to mention getting a solid logical division in the program. The algorithm simply splits each line in the dataset using the split operator "|" when processing
the DBLP dataset, and it uses ";" when processing the Facebook, Facebook Friends, or MovieLens datasets. After that, the algorithm adds the split line content to the variable $DBlines$ to be used later for calculating the similarity between the data. Algorithms 2, 3, and 4 are used to find the Set Similarity between all pairs of data in the selected dataset (Facebook, MovieLens, or DBLP). Algorithms 5, 6, and 7 are used to find the Wordnet-Cosine Similarity between all pairs of data in the selected dataset (Facebook, MovieLens, or DBLP). Algorithms 8, 9, and 10 are used to find the Semantic Category Similarity between all pairs of data in the selected dataset (Facebook, MovieLens, or DBLP). Algorithms 11, 12, and 13 are used to find the Vector Space similarity between all pairs of data in the selected dataset (Facebook, MovieLens, or DBLP).
Algorithm 1: Fill_with_lines_from_dataset

Data: dataset

Result: content of dataset stored in a hashtable called DBlines

/* file to read depends on the dataset that has been specified. */

1 while there are lines in the dataset do
2    if dataset = "Facebook" then
3        split line using ;
4        add first element (user ID) as a key to DBlines
5        add second element (movies interest) as a value to DBlines
6    else
7        if dataset = "MovieLens" then
8            split line using "|
9            add first element (Movies ID) as a key to DBlines
10           add second element (movies title) as a value to DBlines
11        else
12            if dataset = "DBLP" then
13                split line using ";"
14                add first element (ID) as a key to DBlines
15                add second element (publication title) as a value to DBlines
16            else
17                if dataset = "Facebook Friends" then
18                    split line using ";"
19                    add first element (user ID) as a key to DBlines
20                    add second element (movies interest, friends interest, and friends) as a value to DBlines
21            end
22        end
23    end
24 end
Line 5 in Algorithm 2 and line 9 in Algorithm 11 write the results into the text files that will be used later to create a matrix in Matlab, see Section 6.2. Each line in the output file has the following format:

id1-id2: similarity (e.g. 339-336: 0.22950819672131148)

**Algorithm 2: GetResultsFor_Set Similarity**

**Data:** DBlines, see Algorithm 1

**Result:** Text file Set_DS.txt that contains pairwise set similarity

/* Initialization */
1 DBlines = lines from the dataset, by invoking Fill_with_lines_from_dataset, see algorithm 1;
2 tagger: the stanford tagger;

/* body */
3 forall the pair of lines stored in DBlines referenced by LineX and LineY do
4  result = find_Set_WordNetSimilarity(DBlines(LineX), DBlines(LineY), tagger);
5  write results to the file "Set_DS.txt";
6 end

**Algorithm 3: Find_Set_WordNetSimilarity**

**Input:** pair of lines from the Dataset: lineX, and lineY, and a tagger

**Result:** Set similarity between the two lines input

/* Initialization */
1 similarity = 0.0

/* Body */
2 Parents1 = getDistance_Set_VectorFromLine(lineX, tagger)
3 Parents2 = getDistance_Set_VectorFromLine(lineY, tagger)
4 IntersectionNV = Parents1 ∩ Parents2
5 similarity = |IntersectionNV| / |Parents1∪Parents2|
Algorithm 4: GetDistance_Set_VectorFromLine

Input: lineX from the Dataset, tagger

Result: Vector representation of lineX based on the Set Similarity

1. forall the entries (movies entries) in lineX excluding the id do
2.    tag all words in entry using tagger.
3.    forall the tagged words ∈ entry do
4.        if tag of word ∈ (NN, NNS, NNP, NNPS) then
5.                if word is stopword or punctuation then
6.                    skip the word
7.                end
8.                add the parents of the word, based on the hierarchy of concepts using Wordnet,
9.                to ParentsSet
10.           end
11.       end
12. return ParentsSet

Algorithm 5: GetResultsFor_Wordnet-Cosine_Similarity

Data: DBlines, see Algorithm 1

Result: Text file Wordnet_DS.txt that contains pairwise Wordnet-Cosine similarity

/* Initialization */
1. DBlines = lines from the dataset, by invoking Fill_with_lines_from_dataset, see algorithm 1;
2. tagger: the stanford tagger;
/* body */
3. forall the pair of lines stored in DBlines referenced by LineX and LineY do
4.    result = find_WordNetSimilarity(DBlines(LineX), DBlines(LineY), tagger);
5.    write results to the file "Wordnet_DS.txt";
6. end
Algorithm 6: Find_WordnetCosine_WordNetSimilarity

Input: pair of lines from the Dataset: lineX, and lineY, and a tagger

Result: Wordnet-Cosine similarity between the two lines input

/* Initialization */
1 similarity = 0.0

/* Body */
2 distan1 = getDistanceVectorFromLine(lineX, tagger)
3 distan2 = getDistanceVectorFromLine(lineY, tagger)
4 similarity = \frac{distan1 \cdot distan2}{\|distan1\| \|distan2\|}

Algorithm 7: GetDistanceVectorFromLine

Input: lineX from the Dataset, tagger

Result: Vector representation of lineX based on the Wordnet-Cosine Similarity

1 forall the entries (movies entries) in lineX excluding the id do
2     tag all words in entry using tagger.
3     forall the tagged words ∈ entry do
4         if tag of word ∈ (NN, NNS, NNP, NNPS) then
5             if word is stopword or punctuation then
6                 skip the word
7             end
8         end
9         StepsRoot = number of concepts that connect word to entity concept based on
10            the hierarchy of concepts, retrieved using Wordnet
11         add StepsRoot to the vector distance
12     end
13 end
14 return distance
Algorithm 8: GetResultsFor_SemantiCategories_Similarity

**Data:** DBlines, see Algorithm 1

**Result:** Text file SCat_DS.txt that contains pairwise Semantic Categories similarity

/* Initialization */

1 DBlines = lines from the dataset, by invoking Fill_with_lines_from_dataset, see algorithm 1;
2 tagger: the stanford tagger;

/* body */

3 forall the pair of lines stored in DBlines referenced by LineX and LineY do
4    result = find_SemanticCategories_Similarity(DBlines(LineX), DBlines(LineY), tagger);
5    write results to the file "SCat_DS.txt";
6 end

Algorithm 9: Find_SemanticCategories_Similarity

**Input:** pair of lines from the Dataset: lineX, and lineY, and a tagger

**Result:** Semantic Categories similarity between the two lines input

/* Initialization */

1 similarity = 0.0

/* Body */

2 distan1 = getDistanceVectorFromLine(lineX, tagger)
3 distan2 = getDistanceVectorFromLine(lineY, tagger)
4 similarity = \frac{distan1 \cdot distan2}{∥distan1∥∥distan2∥}
**Algorithm 10: GetDistance_SemCat_VectorFromLine**

**Input:** lineX from the Dataset, tagger

**Result:** Vector representation of lineX based on the Semantic Categories Similarity

/* Initialize counters: */

1. \( \text{NN} = 0, \text{NNS} = 0, \text{NNP} = 0, \text{NNPS} = 0 \)

/* body */

2. for all the entries (movies entries) in lineX do

3. tag all words in entry using tagger.

4. for all the tagged words \( \in \) entry do

5. if tag of word \( \in \) (NN, NNS, NNP, NNPS) then

6. if word is stopword or punctuation then

7. skip the word

8. end

9. increment the appropriate counter (e.g. \( \text{NN}++ \) if word is NN)

10. end

11. end

12. end

13. add the counters NN, NNS, NNP, NNPS respectively to the vector ParentSet.

14. return ParentSet.
Algorithm 11: GetResultsFor_VectorSpace_Similarity

**Data:** DBlines, see Algorithm 1

**Result:** Text file Vector_DS.txt that contains pairwise Word Frequency Vector similarity

```plaintext
/* Initialization */
1 DBlines = lines from the dataset, by invoking Fill_with_lines_from_dataset, see algorithm 1;
2 tagger: the stanford tagger;

/* body */
3 forall the words in Dataset that are ∈ (NN, NNS, NNP, NNPS) do
4   store the word along with its number of occurrence in DBWords;
5 end
6 DBwords: contains all words in Dataset and their frequencies (loop through all data);
7 forall the pair of lines stored in DBlines referenced by LineX and LineY do
8   result = find_VectorSpace_Similarity(DBlines(LineX), DBlines(LineY), DBWords);
9   write results to the file Vector_DS.txt;
10 end
```

Algorithm 12: Find_VectorSpace_Similarity

**Input:** pair of lines from the Dataset: lineX, and lineY, DBWords

**Result:** Word Frequency Vector similarity between the two lines input

```plaintext
/* Initialization */
1 similarity = 0.0

/* Body */
2 distan1 = getDistance_VectorSpSim_VectorFromLine(lineX, DBWords)
3 distan2 = getDistance_VectorSpSim_VectorFromLine(lineY, DBWords)
4 similarity = \frac{distan1 \cdot distan2}{\|distan1\| \|distan2\|}
```
Algorithm 13: GetDistance_VectorSpSim_VectorFromLine

Input: lineX from the Dataset, DBwords

Result: Vector representation of lineX based on the Word Frequency Vector Similarity

1. forall the words in DBWords do
2.     initialize Freq = 0
3.     forall the wordX in lineX do
4.         if if wordX is equal to the word in DBWords then
5.             assign the frequency of wordX in lineX to Freq
6.         end
7.     end
8.     if the word in DBWords was found in lineX then
9.         add Freq to wordFrequency
10.     else
11.         add 0 to wordFrequency
12.     end
13. end
14. return wordFrequency ;

5.1.1 Time Complexity Analysis of the Similarity Algorithms

The running time complexities of the similarity algorithms described above, are as follows:

Set similarity: In algorithm 2, the main body of the algorithm is a for-loop that is executed $N \times (N - 1)/2$ times, where $N$ is the number of data instances (lines in the dataset that refer to a user). In each of these iterations, Algorithm 3 is invoked which also invokes Algorithm 4 twice to find the similarity. Algorithm 4 iterates through all of the words in each entry in the passed line (lineX), so its running time complexity equals $E \times W$, where $E$ denotes the number of entries in the line of data (e.g. movies interest), and $W$ is the number of words in one entry. More specifically, $E$ and $W$ are the expected values of the number of entries and words respectively. Therefore, the running time complexity of the Set similarity is: $O(N \times N \times (E \times W))$. 
Wordnet-Cosine and Semantic Categories similarity have the same running time complexity as Set similarity, which is $O(N \times N \times (E \times W))$.

Word Frequency Vector similarity time complexity is $O(N \times N \times (WDB \times WL))$, where $WDB$ is the number of distinctive words in the dataset, and $WL$ is the expected number of words in each line of the dataset.

Table 5.1 summarizes the running time complexity analysis discussed previously.

<table>
<thead>
<tr>
<th>Similarity measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set similarity</td>
<td>$O(N \times N \times (E \times W))$</td>
</tr>
<tr>
<td>Wordnet-Cosine similarity</td>
<td>$O(N \times N \times (E \times W))$</td>
</tr>
<tr>
<td>Semantic Categories similarity</td>
<td>$O(N \times N \times (E \times W))$</td>
</tr>
<tr>
<td>Vector Space similarity</td>
<td>$O(N \times N \times (WDB \times WL))$</td>
</tr>
</tbody>
</table>

5.1.2 Space complexity of the similarity algorithms

The space consumption of all of the previously illustrated algorithms is basically the size of the $DBlines$ and the size of the output text file. The size of the $DBlines$ is $N$, where $N$ is the number of lines in the dataset files. The size of the text output file is $N^2$, because we calculate the similarity between every pair of lines in the datasets.
Chapter 6

Experiments and Results

6.1 Experiments

Several experiments have been conducted in order to evaluate the quality of different similarity measures. In the first set of experiments, clustering was used to group the data (e.g. nodes in profiles in Facebook) into k clusters, with k = 2, 3, 4, 5, and 6 based on the similarity measures. The quality of the clusters was studied to find out which similarity measure produces the best clustering results.

In the second set of experiments, the Wordnet-Cosine similarity measure was used to predict links in a social random graph. The predictor accuracy was compared with the accuracy of a random link predictor.

Facebook, Movielens, and DBLP data were used for the first set of experiments. The information in the MovieLens dataset is shown in Table 6.1 (this information was taken from the read me file of the MovieLens dataset \[^1\]). Only the id and the movie title from the MovieLens were used in finding the similarity between all pairs of data. The id was used to reference the data tuple (row).

Tables 6.2, 6.3, and 6.4 show the first 10 records of each of these data sets respectively. In the Facebook dataset, each row represents a Facebook user identified by an id (auto generated) and his/her movies interest. Table 6.5 displays the size of the distance matrices used in the clustering experiments. These matrices were first compiled by applying the four similarity measures, Cat (Semantic Categories), (Set), Vector (Word Frequency Vector), and Wordnet (Wordnet-Cosine),

\[^1\]http://grouplens.org/datasets/movielens/
Table 6.1: The Format for the Movielens Dataset

<table>
<thead>
<tr>
<th>movie id</th>
<th>movie title</th>
<th>release date</th>
<th>video release date</th>
<th>IMDb URL</th>
<th>unknown</th>
<th>Action</th>
<th>Adventure</th>
<th>Animation</th>
<th>Children’s</th>
<th>Comedy</th>
<th>Crime</th>
<th>Documentary</th>
<th>Drama</th>
<th>Fantasy</th>
<th>Thriller</th>
<th>War</th>
<th>Western</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The last 19 fields are the genres, a 1 indicates the movie is of that genre, a 0 indicates it is not; movies can be in several genres at once.
The movie ids are the ones used in the u.data data set.

Table 6.2: Facebook Dataset: the First 10 Records

1; Harry Potter, Transformers, Mr. & Mrs. Smith,
2; Sherina’s Adventure,
3; Love mein Gum, Maqsood Jutt Dog Fighter,
4; Crows Zero, Detective Conan: Crossroad in the Ancient Capital, miyabi,
5; Charlie and the Chocolate Factory, GARUDA DI DADAKU THE MOVIE, Transformers,
6; Transformers: Revenge of the Fallen, Transformers,
7; Mangaatha, BILLA (), Latest Kollywood News, Vinnaithandi Varuvaaya,
8; Initial D movie, 2012, Fast & Furious, Spider-Man 3,
9; Ketika Cinta Bertasbih 1 Dan 2, Ayat-Ayat Cinta, The Lord of the Rings,
10; The Last Samurai,

discussed previously, on the three datasets. Then the resulted similarity values were assembled into a similarity matrix which was then converted into a distance matrix.

Figures 6.1a, 6.1b, 6.1c, and 6.1d shows the histogram of distances for the four similarity measures applied on Facebook dataset. The distance values are shown on the x-axis, while the y-axis shows the frequency of these values in the distance matrices. The histograms were generated in Matlab using the command "hist(MatrixName(:))". As it can be seen from these figures, the Set distance has the highest granularity (largest number of bins in the histogram, which is equal to 8), while both the Semantic Cat distance and the Wordnet-Cosine distance have smaller granularity (with 3 bins each).

Figures 6.2a, 6.2b, 6.2c, and 6.2d display the histograms of different distances measure applied on the MovieLens dataset. The same pattern of histogram is extended for each similarity measure.
Figure 6.1: Histogram of distances of Facebook data for the four similarity measures
Figure 6.2: Histogram of Distances of MovieLens data for the four Similarity Measures
Figures 6.3a, 6.3b, 6.3c, and 6.3d display the histogram of the distance of the four distance applied on the DBLP data. The same pattern applies where the set similarity has the highest granularity (8 bins), followed by Semantic Categories and Vector Frequency distance (4 bins), and finally the Wordnet-Cosine distance (with the lowest granularity).

The Image plot of the four data matrices: D\textsubscript{Cat,FD,N,F}, D\textsubscript{Set,FD,N,F}, D\textsubscript{Vector,FD,N,F}, and D\textsubscript{WordnetCosine,FD,N,F} are shown in figure 6.4a, 6.4b, 6.4c, and 6.4d. These data matrices include the pair-wise distances of the Facebook dataset using the four measures (Semantic Category, Set, Word Frequency Vector, and Wordnet-Cosine). As it can be seen from the plots, the Wordnet-Cosine data matrix has the largest number of high distances followed by Semantic Categories.

Word Frequency Vector distance matrix has the lowest distance values. Table 6.6 shows some characteristics of these data matrices. A detailed description of how these matrices were created is included in section 6.2.

(a) Semantic Categories data matrix  
(b) Set distance matrix  
(c) Word Frequency Vector matrix  
(d) Wordnet-Cosine data matrix

Figure 6.4: The Image Plots of Facebook Distance Matrices

6.2 Creating the similarity and Distance Matrices

The Facebook dataset, the MovieLens dataset, and the DBLP dataset were clustered using the four distance/similarity measures, and then the average silhouette values were calculated for clustering using k = 1-6. The four similarity measures, studied here, were implemented using Java. Algorithm 14 shows how one similarity measure (Semantic Categories) was used to create similarity matrix in
Matlab from the Facebook dataset. This process can be described as follows:

- An output text file is created that stores the similarity between every pair of users and that resembles a line in the output text file. Algorithm 14 reads the similarity values for all pairs of data (all lines in the text file) and converts them into a similarity matrix so that they can be clustered and explored in depth.

- Algorithm 15 displays the code for converting the similarity matrix into distance matrix (replacing the NaN values), clustering, and finding the average silhouette using Matlab. The similarity matrix (the "SM" prefix in the matrix name) was converted to distance matrix (the D prefix in the matrix name) by using the equation:

\[ d(x, y) = \frac{1}{s(x, y)} - 1 \]

(references [14], see algorithm 15 line 1. That means:

- when \( s(x, y) = 0 \) \( \rightarrow d(x, y) = \infty \) (dissimilar)
- when \( s(x, y) = 1 \) \( \rightarrow d(x, y) = 0 \) (similar).

- The NaN values are replaced by maximum distance, in what is known as *Imputation* [56], see algorithm 15 lines 2-9
Algorithm 14: Algorithm to create a Similarity Matrix

**Data:** the file output from java code, (see Chapter 5 Algorithms 2, 5, 8 and 11)

**Result:** a similarity matrix SM_Cat_FD

1. initialize the (585,585) matrix SM_Cat_FD to 0s ;
2. while not at end of ¨..585_Dataset_semanticCat.txt¨ do
   3. tline = next line from file ;
   4. contentline = split "tline" using ":" ;
   5. ids = split "contentline{1}" using ":" ;
   6. SM_Cat_FD(ids{1}, ids{2}) = contentline{2} ;
7. end

Algorithm 15: Algorithm to create the Distance Matrices from of the Similarity Matrices produced using Algorithm 14

**Data:** a similarity matrix SM_Cat_FD

**Result:** a distance matrix D_Cat_FD_N_F

1. initialization: D_Cat_FD_N_F = 1./SM_Cat_FD - 1;
2. Find the maximum element in D_Cat_FD_N_F that is not NaN or infinity (MaxC). ;
3. forall the elements of D_Cat_FD_N_F referenced by the indices i and j do
   4. if isnan(D_Cat_FD_N_F(i,j)) then
      5. D_Cat_FD_N_F(i,j) = MaxC;
   6. end
7. end
8. cluster the data in the distance matrix D_Cat_FD_N_F into different number of clusters (2,3,4,5,6) using KernelKmeans which is a modified version of kmeans that takes as an input a distance matrix ;
9. find the average silhouette for clustering using k = 1-6 (see the Appendix Section 8.1.2);
6.3 Clustering using Similarity and the Silhouette Index for Clustering Validation

The results obtained by finding the similarity between the movies in the Facebook dataset, Facebook Dataset.txt, were used to find the distance matrix which was then used to cluster the data. This was also repeated for the Movielen and DBLP datasets.

6.3.1 Kmeans Clustering

Kmeans is a method for clustering data, and it is supported by Matlab (kmeans function) [24]. Clustering refers to the problem of grouping the data points into sets which are as tight/compact and as separated as possible. Clustering is used as a ground truth to evaluate the similarity/distance measures since we assume that similar profiles will be in the same cluster when one attempts to cluster them using the similarity/distance measures. Therefore, after clustering the data using the various distance measures, we study the quality of clusters using the Silhouette index to find out which similarity/distance measure cluster the data the best. In this study, a modified version of Kmeans has been used (see the Appendix Section 8.1.3).

6.3.2 Silhouette

The silhouette index has been used to measure the quality of clusters for each of the four similarity measures. The Silhouette index captures the compactness and separation of clusters and is defined as follows (Equation 6.2 and 6.1) [51]:

\[
S(i) = \begin{cases} 
1 - \frac{a(i)}{b(i)} & \text{if } a(i) < b(i) \\
0 & \text{if } a(i) = b(i) \\
\frac{b(i)}{a(i)} - 1 & \text{if } a(i) > b(i)
\end{cases} \tag{6.1}
\]

\[
S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{6.2}
\]

Where \(-1 \leq S(i) \leq 1\), and \(i\) denotes data point and the quantities \(b(i)\) and \(a(i)\) are defined as follows:
• $a(i)$ is the average disimilarity of $i$ to all other data items in the same cluster that $i$ has been assigned to.

• $b(i)$ is the minimum average disimilarity to the clusters to which $i$ doesn’t belong.

In the following the k-mean clustering of the Facebook dataset using the four similarity measures, discussed previously, is detailed followed by the average silhouette values.

6.3.3 Facebook

The Facebook dataset was clustered using the modified clustering algorithm, Kernelkmean, which takes as an input a distance matrix.

Set Distance

Figure 6.5 shows the results of clustering the Facebook dataset using the Set distance. As it can be noticed from the figures, increasing the number of clusters tends to capture more differences between the data (data belongs to different clusters rather than few clusters).

Wordnet-Cosine Distance

Figures 6.6 show the results of clustering the Facebook dataset using the Wordnet-Cosine distance. As it can be seen from the figures, the data spread out more when using the Set distance (see Figure 6.5).

Semantic Categories Distance

Figures 6.7 shows the results of clustering the Facebook dataset using the Semantic Categories distance.

Word Frequency Vector Distance

Figures 6.8 shows the results of clustering the Facebook dataset using the Vector distance.
**Average Silhouette**

Figures 6.9 shows the average silhouette for clustering the Facebook dataset into $k=1$-6 clusters using Set, Wordnet-Cosine, Semantic Categories, and Vector distances respectively. As it can be seen from Figure 6.9, the Vector distance has the highest clustering quality (silhouette value of almost 1) followed by Wordnet-Cosine and Cat distance. The Set distance seems more random than the other similarity. More results can be found by varying the number of clusters.

![Figure 6.5: Clustering using Set Distance applied on Facebook Dataset](image)

![Figure 6.6: Clustering using Wordnet-Cosine Distance applied on Facebook Dataset](image)

![Figure 6.7: Clustering using Semantic Categories Distance applied on Facebook Dataset](image)

### 6.3.4 DBLP

The DBLP dataset was also clustered using Kernelkmean. After that, the silhouette index, obtained from the clustering results of each similarity/distance measure, was calculated to evaluate how well each measure clusters the data.
Figure 6.8: Clustering using Word Frequency Vector Distance applied on Facebook Dataset

(a) 2 clusters   (b) 3 clusters   (c) 4 clusters   (d) 5 clusters   (e) 6 clusters

Figure 6.9: Average Silhouette for clustering the Facebook Dataset into 1-6 Clusters using different Distance Measures

(a) Set   (b) Wordnet-Cosine   (c) Cat   (d) Vector

Set Distance

Figure 6.10 shows the results of clustering the DBLP dataset using the Set distance. Only when using \( k = 5 \) and 6 clusters it can be noticed that more clusters are being used to divide the data into groups.

Wordnet-Cosine Distance

Figures 6.11 show the results of clustering the DBLP dataset using the Wordnet-Cosine distance. In contrast to the results of applying the Set distance on the DBLP dataset, it can be clearly seen that more than one cluster was used to group the data. The number of clusters in the data increases with increasing values for \( k \). More specifically, when \( k=4,5 \) and 6, the results shows that several clusters were detected in the data.

Semantic Categories Distance

Figures 6.12 shows the results of clustering the DBLP dataset using the Semantic Categories distance. Even though the results show the data as condensed, the modified version of kmeans was able to find different clusters in the data.
Word Frequency Vector Distance

Figures 6.13 shows the results of clustering the DBLP dataset using the Vector distance.

Average Silhouette

Figures 6.14 shows the average silhouette for clustering the DBLP dataset into k=1-6 clusters using Set, Wordnet-Cosine, Semantic Categories, and Vector distances respectively. As it can be seen from figure 6.9, the Set distance has the highest silhouette value (1) for k = 2 to 4, followed by Wordnet-Cosine (0.9).

Figure 6.10: Clustering using Set distance applied on DBLP Dataset

Figure 6.11: Clustering using Wordnet-Cosine Distance applied on DBLP Dataset

Figure 6.12: Clustering using Semantic Categories Distance applied on DBLP Dataset
6.3.5 Movielens

The Movielens dataset was also clustered using the modified clustering algorithm, kernelkmean, then the average silhouette was found for each distance/similarity measure.

**Set Distance**

Figure 6.15 shows the results of clustering the Movielens dataset using the Set distance. As $k = 2$, and 3, most of the data belongs to one cluster (cluster 1 when $k = 3$, and cluster 3 when $k = 4$). However, when $k = 5$, and 6 the data spreads to different clusters.

**Wordnet-Cosine Distance**

Figures 6.16 show the results of clustering the Movielens dataset using the Wordnet-Cosine distance.

**Semantic Categories Distance**

Figures 6.17 shows the results of clustering the Movielens dataset using the Semantic Categories distance.
Word Frequency Vector Distance

Figures 6.18 shows the results of clustering the Movielens dataset using the Vector distance.

Average Silhouette

Figures 6.19 shows the average silhouette for clustering the Movielens dataset into k=1-6 clusters using Set, Wordnet-Cosine, Semantic Categories, and Vector distances respectively. As it can be seen from figure 6.19, Wordnet-Cosine distance has the highest silhouette for k=1-6 (avg silhouette equals to 1), while the other distance measures have non-increasing nor decreasing average silhouette (random). This shows that Wordnet-Cosine distance performance is relatively better than that of other similarity measure when applied on some dataset (Movielens), therefore it can be used as a distance/similarity measure.

Figure 6.15: Clustering using Set distance applied on Movielens Dataset

Figure 6.16: Clustering using Wordnet-Cosine distance applied on Movielens Dataset

Figure 6.17: Clustering using Semantic Categories Distance applied on Movielens Dataset
6.4 Comparison between Different Measures

Recall that for two sets, $A$ and $B$, the Jaccard similarity between them is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

The Set Similarity is a Jaccard-like similarity between sets, where $A$ is the set of friends for user $M$, and $B$ is the set of friends for user $N$ ($M$, and $N$ are represented by their profiles). The difference between Jaccard similarity and Set Similarity is as follows: the set similarity computes the common feature values over the number of feature values for user $M$ or $N$. On the other hand, the Jaccard similarity considers the friends of a user instead, and it computes the number of common feature values for the friends of both user $A$ and $B$ divided by the number of distinct feature values for the friends of the first or the second user. This is quite similar to the Jaccard Coefficient [18] which is equal to the number of common friends divided by the number of all friends for either user $A$ or user $B$.

As it has been mentioned previously, the Facebook dataset contains the users ids and their movies interest. Table 6.7 is snapshot of the extended Facebook dataset which contains the friends home pages and their movies interests in addition to the user id and movies interests for Facebook
Table 6.8 shows the statistics of the extended Facebook dataset. As it can be noticed, the Extended Facebook Dataset exhibits almost the same characteristic as the Facebook dataset. Also, even though the set of Facebook users were selected from an arbitrary list (Skullsecurity in contrast to crawl a graph from Facebook), that was not reflected in the users’ friends in the sense that there are friends which are shared between some users as the table indicates (number of friends Distinct Friends Count > 0). This probably suggests that the gathered synthetic dataset might accordingly exhibit some characteristics as a crawled data graph.

Table 6.9 shows the common movies interest between the friends of two Facebook users and the values of different similarity measures. As it can be seen from the table, the Semantic Categories gives the highest similarity value (0.9), where Wordnet-Cosine and Vector gives a similarity value around 0.7. The least similarity is given by Set similarity (0.034). This proves that Wordnet-Cosine similarity is not extreme in the sense it doesn’t give high similarity value for users whose friends don’t have many movies in common (in this example, the common movies interest between the friends of users identified by id 305 and 389 are only two: Titanic, and Twilight).

Table 6.10 shows another comparison between the similarity measures for two Facebook profiles along with more details about the profiles such as: shared friends (shared), and shared/common movies between friends (common). As the table indicates, the Wordnet-Cosine similarity doesn’t give a high similarity for the profiles identified by the ids 441 and 134, and that is due to not having shared movies interest between the two users, while for the same pair of profiles, the Semantic Categories similarity gives the highest similarity measure. Our assumption is that edges (friends connection) in the graph, which represents the extended Facebook dataset, may induce a similarity between the two users that are connected. As it been mentioned previously, Jaccard similarity between user 441 and 134 calculates the similarity as the intersection of movies interest between the friends of profile 441 and profile 134 divided by the union of movies interest between the friends of the two profiles.
6.5 Correlation between the Similarity Measures

Correlation is a relationship between two random variables where changing one variable will result in a change to the other variable. It can be negative or positive indicating the type of correlation between the two variables [10]. Table 6.11 shows the correlation between the different similarity measures considered.

The correlations between the following measures were calculated (see Table 6.11): Wordnet-Cosine (WC), WordFrequencyVector (WFV or Vector), Set, SemanticCat (SC), Jaccard (between the interests of friends), the number of common interest between friends (common), and the number of shared friends (shared). As a reminder, it must be stated that correlation does not imply causation [3].

As it can be seen from the Table 6.11, the highest correlation is between the number of common interest between friends and Jaccard, followed by the correlation between Shared and Jaccard, which enforces the intuition that more shared friends, will results in high number of shared movies interest between the friends of two Facebook users. More specifically, this can be written as: an increase in the number of shared friends leads to an increase in the number of shared movies interest between Facebook users. Moreover, the correlation between shared and common is considered relatively quite high.

Figure 6.20 demonstrates the scatter plot for the number of common movies interest between friends of users and the Jaccard similarity.

6.6 Link Prediction using the Wordnet-Cosine similarity/distance measure

Using the Gelphi software [4] the random graph, as shown in Figure 6.21, was generated with the following properties:

- number of nodes: 585
- number of edges: 892

Figure 6.20: Correlation between the Number of Common Friends’ Interests and Jaccard

- wiring (the proportion of edges over all possible edges given 585 nodes \((892/(585 \times 584)/2)\): 0.005

- Average Clustering Coefficient (a measure that quantifies how the nodes in the graph tend to be in the same cluster): 0.003

- Diameter (the maximum distance between all connected vertices): 12

- Average Path length (the average of the \(585 \times 584/2\) path lengths): 5.6334649433773265

- Number of shortest paths: 307474

The Wordnet-Cosine distance measure for link prediction was tested against a random method as follows:

(1) Randomly delete a percentage (10%, 20%, etc.) of links from the network

Different percentages of links were deleted. In Table 6.12, 20% of links were deleted randomly. More specifically, \(|(0.2 \times 892) = 178|\) edges are deleted from the random graph. In Table 6.13, different percentages, [10% - 90%], of links were deleted.
(2) Test how many of the deleted links are reinstated by using the distance measure obtained from the Wordnet-Cosine

(a) A link was predicted using the distance measure if it has a distance value < $\alpha$, where $\alpha = 0.5$ in the example Table 6.12. Different values for $\alpha$ were experimented with as it is shown in Table 6.13.

(b) Another matrix which gives the distance between all pair of users was used to predict links, alongside the random graph adjacency matrix. This matrix contains the Wordnet-Cosine distance value for all pair of 585 users in the Facebook dataset. We assumed that the Wordnet-cosine distance matrix contains movies interest of the nodes in the random graph. In other words, we have created a synthetic data that contains a random graph.
(generated by Gelphi) and movies interest of different users (generated by Skullsecurity list).

(3) Evaluate the number of links reinstated by a random procedure

A random binary vector, \( x \), of \{0, 1\} values, was generated with a size equal to the number of deleted edges, that is, here the random vector of dimension 178 of 1s/0s to mean that a link is predicted/not predicted.

(4) Evaluate how different are the results from (2) and (3).

Calculate the number of predicted links using (2) and (3): The exclusive-or operator is applied to the two adjacency matrices: for the prediction based on distance and random, and the random graph after deleting from it the percentage of links, mentioned above.

From Table 6.12 it can be concluded that the Wordnet-cosine distance outperformed the random link prediction method by a factor of \( \frac{117 - 90}{117} \times 100 = 23.07\% \) based on the experiment conducted. Furthermore, it can be seen that the Wordnet-Cosine distance was able to predict \( \frac{117 - 178}{100} = 65.73\% \) of deleted edges (links). Based on the detailed experiment above, the accuracy of predicting links using Random is \( \frac{90}{178} \times 100 = 50.56\% \).

The accuracy of random link prediction and Wordnet-Cosine distance link prediction, using different values for deleted edges and distance threshold, is shown in the Table 6.13. The table shows the number of deleted edges from the random graph (edges del), the percentage of deleted edges (% edges del), the Distance Threshold (Dist T), the number of predicted links using Random (predicted Ran), the number of predicted links using Wordnet-Cosine distance (predicted Dist), the Random predictor accuracy (Rand Acy), and the Wordnet-Cosine predictor accuracy (dist Acy).

For the Wordnet-Cosine distance link prediction, a link was predicted if the distance between the two nodes is less than the Distance Threshold (Dist T). The accuracy of the random predictor ranges from 37.08\% to 58.43\%, while the accuracy of the Wordnet-Cosine distance predictor ranges from 0.056 to 65.73\%. Which means that Wordnet-Cosine distance-based predictor achieves higher accuracy than the random predictor (65.73\% v.s. 58.43\%). See the probability of correctness for the random predictor [35]. As it can be seen from Table 6.13 the Wordnet-Cosine distance link predictor, has a higher accuracy for most cases where the Distance Threshold (Dist T) =
2 to 10. However, when the Dist $T = 1$, the random link predictor accuracy is larger than the Wordnet-Cosine accuracy for all values of deleted edges (edges del/ % edges del).

However, an issue to investigate is the relationship, if any, between the Random predictor accuracy and the Wordnet-Cosine distance predictor accuracy. To this end, the correlation coefficient between these two predictions is found to be $\rho = -0.15$. Based on the sample size of 108 obtained by considering all combinations of percentages of deleted edge pairs and distance thresholds (see Table 6.13), the $p$-value for the statistical test

$$H_0 : \rho = 0 \text{ versus } H_1 : \rho \neq 0$$

is found to be $0.06 < p < 0.07$. Since $p > 0.05$ the null hypothesis, $H_0$ (that the two predictors are uncorrelated) cannot rejected. In other words, it can be concluded, that the random accuracy and the Wordnet-Cosine accuracy are uncorrelated.

Table 6.13: Accuracy of Wordnet-Cosine and Random Predictors

<table>
<thead>
<tr>
<th>edges del</th>
<th>% edges del</th>
<th>Dist T</th>
<th>predicted Ran</th>
<th>predicted Dist</th>
<th>Rand Acy</th>
<th>dist Acy</th>
</tr>
</thead>
<tbody>
<tr>
<td>89</td>
<td>0.1</td>
<td>10</td>
<td>52</td>
<td>47</td>
<td>58.43</td>
<td>52.81</td>
</tr>
<tr>
<td>89</td>
<td>0.1</td>
<td>9</td>
<td>33</td>
<td>47</td>
<td>37.08</td>
<td>52.81</td>
</tr>
<tr>
<td>89</td>
<td>0.1</td>
<td>8</td>
<td>39</td>
<td>47</td>
<td>43.82</td>
<td>52.81</td>
</tr>
<tr>
<td>89</td>
<td>0.1</td>
<td>7</td>
<td>40</td>
<td>47</td>
<td>44.94</td>
<td>52.81</td>
</tr>
<tr>
<td>89</td>
<td>0.1</td>
<td>6</td>
<td>48</td>
<td>47</td>
<td>53.93</td>
<td>52.81</td>
</tr>
<tr>
<td>89</td>
<td>0.1</td>
<td>5</td>
<td>45</td>
<td>47</td>
<td>50.56</td>
<td>52.81</td>
</tr>
<tr>
<td>89</td>
<td>0.1</td>
<td>4</td>
<td>39</td>
<td>45</td>
<td>43.82</td>
<td>50.56</td>
</tr>
<tr>
<td>89</td>
<td>0.1</td>
<td>3</td>
<td>52</td>
<td>43</td>
<td>58.43</td>
<td>48.31</td>
</tr>
<tr>
<td>89</td>
<td>0.1</td>
<td>2</td>
<td>45</td>
<td>40</td>
<td>50.56</td>
<td>44.94</td>
</tr>
<tr>
<td>89</td>
<td>0.1</td>
<td>1</td>
<td>39</td>
<td>28</td>
<td>43.82</td>
<td>31.46</td>
</tr>
<tr>
<td>89</td>
<td>0.1</td>
<td>0.5</td>
<td>42</td>
<td>13</td>
<td>47.19</td>
<td>14.61</td>
</tr>
<tr>
<td>89</td>
<td>0.1</td>
<td>0</td>
<td>49</td>
<td>1</td>
<td>55.06</td>
<td>1.12</td>
</tr>
<tr>
<td>178</td>
<td>0.2</td>
<td>10</td>
<td>90</td>
<td>117</td>
<td>50.56</td>
<td>65.73</td>
</tr>
</tbody>
</table>

Continued on next page
Table 6.13 – continued from previous page

<table>
<thead>
<tr>
<th>edges del</th>
<th>% edges del</th>
<th>Dist T</th>
<th>predicted Ran</th>
<th>predicted Dist</th>
<th>Rand Acy</th>
<th>dist Acy</th>
</tr>
</thead>
<tbody>
<tr>
<td>178</td>
<td>0.2</td>
<td>9</td>
<td>89</td>
<td>117</td>
<td>50.00</td>
<td>65.73</td>
</tr>
<tr>
<td>178</td>
<td>0.2</td>
<td>8</td>
<td>92</td>
<td>117</td>
<td>51.69</td>
<td>65.73</td>
</tr>
<tr>
<td>178</td>
<td>0.2</td>
<td>7</td>
<td>89</td>
<td>117</td>
<td>50.00</td>
<td>65.73</td>
</tr>
<tr>
<td>178</td>
<td>0.2</td>
<td>6</td>
<td>84</td>
<td>117</td>
<td>47.19</td>
<td>65.73</td>
</tr>
<tr>
<td>178</td>
<td>0.2</td>
<td>5</td>
<td>80</td>
<td>116</td>
<td>44.94</td>
<td>65.17</td>
</tr>
<tr>
<td>178</td>
<td>0.2</td>
<td>4</td>
<td>96</td>
<td>113</td>
<td>53.93</td>
<td>63.48</td>
</tr>
<tr>
<td>178</td>
<td>0.2</td>
<td>3</td>
<td>88</td>
<td>109</td>
<td>49.44</td>
<td>61.24</td>
</tr>
<tr>
<td>178</td>
<td>0.2</td>
<td>2</td>
<td>83</td>
<td>100</td>
<td>46.63</td>
<td>56.18</td>
</tr>
<tr>
<td>178</td>
<td>0.2</td>
<td>1</td>
<td>85</td>
<td>62</td>
<td>47.75</td>
<td>34.83</td>
</tr>
<tr>
<td>178</td>
<td>0.2</td>
<td>0.5</td>
<td>87</td>
<td>31</td>
<td>48.88</td>
<td>17.42</td>
</tr>
<tr>
<td>178</td>
<td>0.2</td>
<td>0</td>
<td>94</td>
<td>1</td>
<td>52.81</td>
<td>0.56</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>10</td>
<td>139</td>
<td>151</td>
<td>51.87</td>
<td>56.34</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>9</td>
<td>125</td>
<td>151</td>
<td>46.64</td>
<td>56.34</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>8</td>
<td>127</td>
<td>150</td>
<td>47.39</td>
<td>55.97</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>7</td>
<td>133</td>
<td>149</td>
<td>49.63</td>
<td>55.60</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>6</td>
<td>128</td>
<td>149</td>
<td>47.76</td>
<td>55.60</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>5</td>
<td>135</td>
<td>148</td>
<td>50.37</td>
<td>55.22</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>4</td>
<td>121</td>
<td>145</td>
<td>45.15</td>
<td>54.10</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>3</td>
<td>142</td>
<td>138</td>
<td>52.99</td>
<td>51.49</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>2</td>
<td>140</td>
<td>123</td>
<td>52.24</td>
<td>45.90</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>1</td>
<td>137</td>
<td>72</td>
<td>51.12</td>
<td>26.87</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>0.5</td>
<td>136</td>
<td>33</td>
<td>50.75</td>
<td>12.31</td>
</tr>
<tr>
<td>268</td>
<td>0.3</td>
<td>0</td>
<td>135</td>
<td>4</td>
<td>50.37</td>
<td>1.49</td>
</tr>
<tr>
<td>357</td>
<td>0.4</td>
<td>10</td>
<td>183</td>
<td>209</td>
<td>51.26</td>
<td>58.54</td>
</tr>
<tr>
<td>357</td>
<td>0.4</td>
<td>9</td>
<td>175</td>
<td>209</td>
<td>49.02</td>
<td>58.54</td>
</tr>
<tr>
<td>357</td>
<td>0.4</td>
<td>8</td>
<td>181</td>
<td>208</td>
<td>50.70</td>
<td>58.26</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>edges del</th>
<th>% edges del</th>
<th>Dist T</th>
<th>predicted Ran</th>
<th>predicted Dist</th>
<th>Rand Acy</th>
<th>dist Acy</th>
</tr>
</thead>
<tbody>
<tr>
<td>357</td>
<td>0.4</td>
<td>7</td>
<td>170</td>
<td>207</td>
<td>47.62</td>
<td>57.98</td>
</tr>
<tr>
<td>357</td>
<td>0.4</td>
<td>6</td>
<td>156</td>
<td>206</td>
<td>43.70</td>
<td>57.70</td>
</tr>
<tr>
<td>357</td>
<td>0.4</td>
<td>5</td>
<td>202</td>
<td>203</td>
<td>56.58</td>
<td>56.86</td>
</tr>
<tr>
<td>357</td>
<td>0.4</td>
<td>4</td>
<td>171</td>
<td>202</td>
<td>47.90</td>
<td>56.58</td>
</tr>
<tr>
<td>357</td>
<td>0.4</td>
<td>3</td>
<td>171</td>
<td>195</td>
<td>47.90</td>
<td>54.62</td>
</tr>
<tr>
<td>357</td>
<td>0.4</td>
<td>2</td>
<td>170</td>
<td>173</td>
<td>47.62</td>
<td>48.46</td>
</tr>
<tr>
<td>357</td>
<td>0.4</td>
<td>1</td>
<td>186</td>
<td>109</td>
<td>52.10</td>
<td>30.53</td>
</tr>
<tr>
<td>357</td>
<td>0.4</td>
<td>0.5</td>
<td>181</td>
<td>48</td>
<td>50.70</td>
<td>13.45</td>
</tr>
<tr>
<td>357</td>
<td>0.4</td>
<td>0</td>
<td>173</td>
<td>2</td>
<td>48.46</td>
<td>0.56</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>10</td>
<td>223</td>
<td>261</td>
<td>50.00</td>
<td>58.52</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>9</td>
<td>217</td>
<td>261</td>
<td>48.65</td>
<td>58.52</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>8</td>
<td>225</td>
<td>261</td>
<td>50.45</td>
<td>58.52</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>7</td>
<td>232</td>
<td>260</td>
<td>52.02</td>
<td>58.30</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>6</td>
<td>229</td>
<td>258</td>
<td>51.35</td>
<td>57.85</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>5</td>
<td>226</td>
<td>254</td>
<td>50.67</td>
<td>56.95</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>4</td>
<td>215</td>
<td>248</td>
<td>48.21</td>
<td>55.61</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>3</td>
<td>244</td>
<td>239</td>
<td>54.71</td>
<td>53.59</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>2</td>
<td>233</td>
<td>215</td>
<td>52.24</td>
<td>48.21</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>1</td>
<td>221</td>
<td>141</td>
<td>49.55</td>
<td>31.61</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>0.5</td>
<td>215</td>
<td>69</td>
<td>48.21</td>
<td>15.47</td>
</tr>
<tr>
<td>446</td>
<td>0.5</td>
<td>0</td>
<td>243</td>
<td>4</td>
<td>54.48</td>
<td>0.90</td>
</tr>
<tr>
<td>535</td>
<td>0.6</td>
<td>10</td>
<td>257</td>
<td>319</td>
<td>48.04</td>
<td>59.63</td>
</tr>
<tr>
<td>535</td>
<td>0.6</td>
<td>9</td>
<td>264</td>
<td>319</td>
<td>49.35</td>
<td>59.63</td>
</tr>
<tr>
<td>535</td>
<td>0.6</td>
<td>8</td>
<td>262</td>
<td>318</td>
<td>48.97</td>
<td>59.44</td>
</tr>
<tr>
<td>535</td>
<td>0.6</td>
<td>7</td>
<td>269</td>
<td>316</td>
<td>50.28</td>
<td>59.07</td>
</tr>
<tr>
<td>535</td>
<td>0.6</td>
<td>6</td>
<td>266</td>
<td>314</td>
<td>49.72</td>
<td>58.69</td>
</tr>
</tbody>
</table>

Continued on next page
Table 6.13 – continued from previous page

<table>
<thead>
<tr>
<th>edges del</th>
<th>% edges del</th>
<th>Dist T</th>
<th>predicted Ran</th>
<th>predicted Dist</th>
<th>Rand Acy</th>
<th>dist Acy</th>
</tr>
</thead>
<tbody>
<tr>
<td>535</td>
<td>0.6</td>
<td>5</td>
<td>277</td>
<td>307</td>
<td>51.78</td>
<td>57.38</td>
</tr>
<tr>
<td>535</td>
<td>0.6</td>
<td>4</td>
<td>274</td>
<td>301</td>
<td>51.21</td>
<td>56.26</td>
</tr>
<tr>
<td>535</td>
<td>0.6</td>
<td>3</td>
<td>269</td>
<td>288</td>
<td>50.28</td>
<td>53.83</td>
</tr>
<tr>
<td>535</td>
<td>0.6</td>
<td>2</td>
<td>284</td>
<td>251</td>
<td>53.08</td>
<td>46.92</td>
</tr>
<tr>
<td>535</td>
<td>0.6</td>
<td>1</td>
<td>252</td>
<td>168</td>
<td>47.10</td>
<td>31.40</td>
</tr>
<tr>
<td>535</td>
<td>0.6</td>
<td>0.5</td>
<td>266</td>
<td>77</td>
<td>49.72</td>
<td>14.39</td>
</tr>
<tr>
<td>535</td>
<td>0.6</td>
<td>0</td>
<td>265</td>
<td>9</td>
<td>49.53</td>
<td>1.68</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>10</td>
<td>306</td>
<td>362</td>
<td>49.04</td>
<td>58.01</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>9</td>
<td>300</td>
<td>362</td>
<td>48.08</td>
<td>58.01</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>8</td>
<td>303</td>
<td>361</td>
<td>48.56</td>
<td>57.85</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>7</td>
<td>311</td>
<td>359</td>
<td>49.84</td>
<td>57.53</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>6</td>
<td>336</td>
<td>356</td>
<td>53.85</td>
<td>57.05</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>5</td>
<td>312</td>
<td>350</td>
<td>50.00</td>
<td>56.09</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>4</td>
<td>308</td>
<td>342</td>
<td>49.36</td>
<td>54.81</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>3</td>
<td>287</td>
<td>327</td>
<td>45.99</td>
<td>52.40</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>2</td>
<td>304</td>
<td>290</td>
<td>48.72</td>
<td>46.47</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>1</td>
<td>323</td>
<td>193</td>
<td>51.76</td>
<td>30.93</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>0.5</td>
<td>329</td>
<td>8</td>
<td>52.72</td>
<td>14.26</td>
</tr>
<tr>
<td>624</td>
<td>0.7</td>
<td>0</td>
<td>319</td>
<td>8</td>
<td>51.12</td>
<td>1.28</td>
</tr>
<tr>
<td>714</td>
<td>0.8</td>
<td>10</td>
<td>337</td>
<td>413</td>
<td>47.20</td>
<td>57.84</td>
</tr>
<tr>
<td>714</td>
<td>0.8</td>
<td>9</td>
<td>354</td>
<td>413</td>
<td>49.58</td>
<td>57.84</td>
</tr>
<tr>
<td>714</td>
<td>0.8</td>
<td>8</td>
<td>369</td>
<td>412</td>
<td>51.68</td>
<td>57.70</td>
</tr>
<tr>
<td>714</td>
<td>0.8</td>
<td>7</td>
<td>347</td>
<td>410</td>
<td>48.60</td>
<td>57.42</td>
</tr>
<tr>
<td>714</td>
<td>0.8</td>
<td>6</td>
<td>345</td>
<td>408</td>
<td>48.32</td>
<td>57.14</td>
</tr>
<tr>
<td>714</td>
<td>0.8</td>
<td>5</td>
<td>361</td>
<td>400</td>
<td>50.56</td>
<td>56.02</td>
</tr>
<tr>
<td>714</td>
<td>0.8</td>
<td>4</td>
<td>346</td>
<td>391</td>
<td>48.46</td>
<td>54.76</td>
</tr>
</tbody>
</table>

Continued on next page
When considering accuracy levels, only 3.7% (4 out of 108) of random accuracies are equal or exceed 55%, and none of them are equal or exceed 59%. By contrast, for the Wordnet-Cosine distance accuracy, 50.9% (55 out of 108) are equal or exceed 55% accuracy, and 11.11% (12 out of 108) are equal or exceed 59%. The distribution of accuracies of the two predictors is further illustrated in Figure 6.22.

Figures 6.23 and 6.24 display the surface plots of the accuracy of the Wordnet-Cosine distance predictor and the random predictor as a function of the Distance Threshold (Dist T) and the Deleted edges (edges del). Figure 6.25 shows the surface plot of the difference between the two predicted surfaces. As it can be seen from Figure 6.23, the accuracy of the Wordnet-Cosine distance
Figure 6.22: Histogram of accuracies for the Random and Wordnet-Cosine distance predictors.

The accuracy for the random predictor increases with increasing the Distance Threshold (Dist T) (predict link \( e_{i,j} \) if \( dist_{i,j} < \alpha \)). However, as it can be noted, the Distance Threshold has much more impact on the accuracy than the percentage of deleted edges, thus the more drop in accuracy as we move right along the Distance Threshold axis (the accuracy drops from 60 to 30).

Figure 6.23: Accuracy of Wordnet-Cosine Distance Predictor.

As it can be seen from Figure 6.24, the accuracy for the random predictor is indeed random (it doesn’t increase nor decrease with the percentage of deleted edges or the Distance Threshold.

75
Figure 6.24: Accuracy of Random Predictor.

Figure 6.25 shows the surface area plot for the difference between the accuracy of Wordnet-Cosine distance and the random predictor (Wordnet-Cosine accuracy − Random accuracy).

It was found that when Distance Threshold (Dist T) = 60 (which is the largest value of distance in the Wordnet-Cosine distance matrix, see 6.4), the accuracy of Wordnet-Cosine distance predictor is 100 (except when %edges_deleted = 0.1, it is equal to 0 in this case), while the accuracy of a random predictor ranges from 48.17 to 53.37, see Table 6.14. See the Appendix section 8.1.4 for the Matlab code for this experiment.

6.6.1 Wordnet-Cosine and Random Graph Adjacency

In another experiment to see how the Wordnet-Cosine distance matrix is similar to the random graph adjacency matrix, the Wordnet-Cosine distance matrix was modified as follows. For distance threshold $\alpha$, and nodes $i$ and $j$, the distance between them, $dist_{i,j}$, is reset as follows:

$$dist_{i,j} \leftarrow \begin{cases} 
0 & \text{if } dist_{i,j} \leq \alpha \\
1 & \text{if } dist_{i,j} > \alpha 
\end{cases}$$

The resulted matrix was used to find the number of different element $Num_{Diff}$ between it and the Random graph adjacency matrix (the number of 1s on the result of applying the XOR on both
Figure 6.25: Difference between the accuracy of Wordnet-Cosine Distance and Random Predictor matrices). The difference between the modified Wordnet-Cosine distance and the Random graph adjacency matrix is shown in Table 6.15. The percentage of difference, $diff\%$

$$diff\% = \frac{Num_{Diff}}{Num_{edges}}$$

where $Num_{edges} = \frac{585 \times 584}{2} = 170,820$.

Figure 6.26: Number of Different Elements between the modified Wordnet-Cosine distance matrix and Random Graph Adjacency matrix

6.7 Conclusion to Chapter 6

Two experiments have been conducted. In the first, the selected similarity/distance measures (Wordnet-Cosine, Semantic Categories, Set, and Vector) were used to cluster three datasets (Facebook, Movielens, and DBLP). After that, the average silhouette was found to assess the clustering quality for each similarity/distance measure.

For the Facebook dataset, the Vector distance was the best followed by Wordnet-Cosine and Cat distance. The Set distance was the least significant in terms of clustering quality of its results. For
the DBLP dataset, the average silhouette for the Set similarity is the highest, followed by that for Wordnet-Cosine (specially when \( k = 2 \)). For the MovieLens dataset, the average silhouette for the Wordnet-Cosine similarity is the highest (equals to 1 for \( k = 2-6 \)). All other similarity measures have what seems to be a random average silhouette (non decreasing, increasing, or constant). Therefore, it can be concluded that Wordnet-Cosine, on average, produces better quality clustering results.

In the second experiment, the Wordnet-Cosine distance was used to predict deleted links from a random graph. The accuracy of this Wordnet-Cosine predictor was compared against the random link predictor. The results confirmed that the Wordnet-Cosine similarity/distance can be used to predict missing links from a random graph as its accuracy increases by increasing the distance threshold (which determines when to predict link).
### Table 6.3: MovieLens Dataset: the First 10 Records

<table>
<thead>
<tr>
<th></th>
<th>Movie Title</th>
<th>Rating</th>
<th>Date</th>
<th>Year</th>
<th>URL</th>
</tr>
</thead>
</table>

### Table 6.4: DBLP Dataset: the First 10 Records

1: Further Normalization of the Data Base Relational Model.  
2: Common Subexpression Identification in General Algebraic Systems.  
3: Principles of Distributed Object Database Languages.  
4: Interactive Support for Non-Programmers: The Relational and Network Approaches.  
5: ACM SIGMOD Contribution Award 2003 Acceptance Speech  
6: ROSAR - Rule Oriented System for Analysis of Reflections on Printed Circuit Boards  
7: Data Base Sublanguage Founded on the Relational Calculus.  
8: Relational Completeness of Data Base Sublanguages.  
9: Derivability, Redundancy and Consistency of Relations Stored in Large Data Banks.  
10: DBLP.uni-trier.de: Computer Science Bibliography
Table 6.5: Characteristic of Distance Matrices used for Clustering

<table>
<thead>
<tr>
<th>Dataset 1 Matrix</th>
<th>Facebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix size</td>
<td>$585 \times 585$</td>
</tr>
<tr>
<td>Dataset 2 Matrix</td>
<td>MovieLens</td>
</tr>
<tr>
<td>Matrix size</td>
<td>$1001 \times 1001$</td>
</tr>
<tr>
<td>Dataset 3 Matrix</td>
<td>DBLP</td>
</tr>
<tr>
<td>Matrix Size</td>
<td>$1000 \times 1000$</td>
</tr>
</tbody>
</table>

Table 6.6: Distance Data Matrices from the Facebook Dataset (_FD)

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Size</th>
<th>Min value</th>
<th>Max Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_Cat_FD_N_F</td>
<td>585x585</td>
<td>$0 \ (2.22044604925031e-16)$</td>
<td>50.2445</td>
</tr>
<tr>
<td>D_Set_FD_N_F</td>
<td>585x585</td>
<td>0</td>
<td>136</td>
</tr>
<tr>
<td>D_Vector_FD_N_F</td>
<td>585x585</td>
<td>0</td>
<td>5.0000</td>
</tr>
<tr>
<td>D_WordnetCosine_FD_N_F</td>
<td>585x585</td>
<td>0</td>
<td>53.7357</td>
</tr>
</tbody>
</table>
Table 6.7: Extended Facebook dataset for User whose id is 1

<table>
<thead>
<tr>
<th>ID</th>
<th>Name and Movie List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Harry Potter, Transformers, Mr. &amp; Mrs. Smith, Download Film, :: Free Movie Freak ::, American Pie, Film Lucu Indonesia, Touch My Heart, Filmygyan, Celeb-states.com, Kutipan Kata-Kata Dari Film, The Terminator, arti sahabat, Kungfu Panda, Avatar, Kal Ho Naa Ho, Ice Age (2002 film), Ayat-Ayat Cinta, Horror film, New Police Story, Bokep Pooool, Play On, CARA MENGGANTI WARNA BERANDA FB SESUKA HATI KAMU! BISA PAKE HP, Crow Zero, Transformers, Harry Potter, Horror,</td>
</tr>
<tr>
<td>11</td>
<td>Harry Potter, Air Terjun Pengantin, HEBOH !!! CARA MUDAH MENAMBAH TEMAN DENGAN STATUS ANIMASI UNIK, 100% WORKING, UPDATE STATUS EMOJI MELALUI HP TANPA PERLU iPHONE, BURUAN, Shrek, Toy Story, Horror, Disney, Ketika Cinta Bertasbih, 2012, He is BeaUtifull,</td>
</tr>
<tr>
<td>12</td>
<td>Harry Potter, Transformers, Mr. &amp; Mrs. Smith, Download Film, :: Free Movie Freak ::, American Pie,</td>
</tr>
<tr>
<td>13</td>
<td>Harry Potter, Air Terjun Pengantin, HEBOH !!! CARA MUDAH MENAMBAH TEMAN DENGAN STATUS ANIMASI UNIK, 100% WORKING, UPDATE STATUS EMOJI MELALUI HP TANPA PERLU iPHONE, BURUAN, Shrek, Toy Story, Horror, Disney, Ketika Cinta Bertasbih, 2012, He is BeaUtifull,</td>
</tr>
<tr>
<td>14</td>
<td>Harry Potter, Air Terjun Pengantin, HEBOH !!! CARA MUDAH MENAMBAH TEMAN DENGAN STATUS ANIMASI UNIK, 100% WORKING, UPDATE STATUS EMOJI MELALUI HP TANPA PERLU iPHONE, BURUAN, Shrek, Toy Story, Horror, Disney, Ketika Cinta Bertasbih, 2012, He is BeaUtifull,</td>
</tr>
<tr>
<td>15</td>
<td>Harry Potter, Air Terjun Pengantin, HEBOH !!! CARA MUDAH MENAMBAH TEMAN DENGAN STATUS ANIMASI UNIK, 100% WORKING, UPDATE STATUS EMOJI MELALUI HP TANPA PERLU iPHONE, BURUAN, Shrek, Toy Story, Horror, Disney, Ketika Cinta Bertasbih, 2012, He is BeaUtifull,</td>
</tr>
<tr>
<td>16</td>
<td>Harry Potter, Air Terjun Pengantin, HEBOH !!! CARA MUDAH MENAMBAH TEMAN DENGAN STATUS ANIMASI UNIK, 100% WORKING, UPDATE STATUS EMOJI MELALUI HP TANPA PERLU iPHONE, BURUAN, Shrek, Toy Story, Horror, Disney, Ketika Cinta Bertasbih, 2012, He is BeaUtifull,</td>
</tr>
</tbody>
</table>
Table 6.8: Extended Facebook Dataset Statistics

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum movies entries</td>
<td>8</td>
</tr>
<tr>
<td>Minimum movies entries</td>
<td>1</td>
</tr>
<tr>
<td>Number of Facebook profiles</td>
<td>454</td>
</tr>
<tr>
<td>Number of movies entries</td>
<td>1336</td>
</tr>
<tr>
<td>Average Movies entries per profile</td>
<td>2.0</td>
</tr>
<tr>
<td>Distinct Movies Count</td>
<td>874</td>
</tr>
</tbody>
</table>

**Friends Statistics**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Friends Count</td>
<td>40</td>
</tr>
<tr>
<td>Minimum Friends Count</td>
<td>1</td>
</tr>
<tr>
<td>Number of Friends</td>
<td>7511</td>
</tr>
<tr>
<td>Average Friends Count per profile</td>
<td>16.0</td>
</tr>
<tr>
<td>Distinct Friends Count</td>
<td>6905</td>
</tr>
</tbody>
</table>

**Friends Movies Statistics**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Friends’ Movies Count</td>
<td>117</td>
</tr>
<tr>
<td>Minimum Friends’ Movies Count</td>
<td>0</td>
</tr>
<tr>
<td>Average Friends Movies per profile</td>
<td>32.0</td>
</tr>
<tr>
<td>Distinct Friends’ Movies Count</td>
<td>5795</td>
</tr>
</tbody>
</table>

Table 6.9: Sample results - Common Movies Interest between the Friends of two Facebook Users identified by ids 305 and 389

<table>
<thead>
<tr>
<th>Facebook users ids</th>
<th>305-389</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of common movies interest between freinds</td>
<td>11</td>
</tr>
<tr>
<td>movies intersect between friends</td>
<td>Titanic, Twilight</td>
</tr>
<tr>
<td>number of shared friends</td>
<td>0</td>
</tr>
<tr>
<td>Wordnet-Cosine similarity</td>
<td>0.7353773632614687</td>
</tr>
<tr>
<td>Set similarity</td>
<td>0.03409090909090909</td>
</tr>
<tr>
<td>Vector similarity</td>
<td>0.7579238282385405</td>
</tr>
<tr>
<td>Semantic Categories similarity</td>
<td>0.949157995752499</td>
</tr>
</tbody>
</table>
Table 6.10: More Comparison between the Similarity Measures for two Facebook Profiles identified by ids 441 and 134

<table>
<thead>
<tr>
<th>Shared Friends between users 441 and 134</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>number of shared friends is 3</td>
<td>![Facebook links](<a href="https://www.facebook.com/rahul.solanki85">https://www.facebook.com/rahul.solanki85</a>, <a href="https://www.facebook.com/harshal.kothari.73">https://www.facebook.com/harshal.kothari.73</a>, <a href="https://www.facebook.com/bhuminish">https://www.facebook.com/bhuminish</a>)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shared movies between friends of users 441 and 134</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>number of shared movies between friends of 441 and 134 is 17</td>
<td>Happy Days, Kuch Kuch Hota Hai Official, 3 Idiots Movie, RHTDM, DUNIYADARI, Apocalypto Movie, 3 on a Bed, Style Apna Apna, Deception, I Love You, Man, Jackass, Inception, Surrogates, Innocent Voices, Maho Toyota, MTV Roadies</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Movies interest for users identified by 441 and 134</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook profile 441</td>
<td>Haunted 3d, Saw, Transformers, Pirates of the Caribbean, Mind Hunter, KUNFU PANDA, The Fast and the Furious: Tokyo Drift, mr.bean, Doctor Dolittle, The Kite Runner,</td>
</tr>
<tr>
<td>Facebook profile 134</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Similarity measures between users 441 and 134</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Semantic</td>
<td>0.9000450033752814</td>
</tr>
<tr>
<td>Set</td>
<td>0.13513513513513514</td>
</tr>
<tr>
<td>Word Frequency Vector</td>
<td>0.7283956834350813</td>
</tr>
<tr>
<td>Wordnet-Cosine</td>
<td>0.415159951099092</td>
</tr>
<tr>
<td>Jaccard</td>
<td>0.1111111111111111</td>
</tr>
</tbody>
</table>

Table 6.11: Correlation between Different Measures

<table>
<thead>
<tr>
<th>Correlations with # shared friends (Shared)</th>
<th>Correlations with # of common movies interest between friends (Common)</th>
<th>Correlations with Wordnet-Cosine (WC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared 0.097196506</td>
<td>WC 0.081172206</td>
<td>WC -</td>
</tr>
<tr>
<td>Shared 0.126641222</td>
<td>Common 0.130383954</td>
<td>WC-WFV 0.021503866</td>
</tr>
<tr>
<td>Shared 0.398701381</td>
<td>- Set 0.295613081</td>
<td>WC-Set 0.150186679</td>
</tr>
<tr>
<td>Shared 0.007378228</td>
<td>Common 0.034189233</td>
<td>WC-SC 0.020973</td>
</tr>
<tr>
<td>Shared 0.811495134</td>
<td>- Jaccard 0.855211546</td>
<td>WC-Jaccard 0.094887507</td>
</tr>
<tr>
<td>Shared 0.687358908</td>
<td>- Common</td>
<td>-</td>
</tr>
</tbody>
</table>

83
Table 6.12: Comparison between Wordnet-Cosine and Random Predictors for DistT = 10

<table>
<thead>
<tr>
<th>Random Graph (RG)</th>
<th>Nodes: 585 Edges:892</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of edges Remaining from (RG)</td>
<td>714</td>
</tr>
<tr>
<td>Number of edges deleted from the (RG)</td>
<td>892 − 714 = 178 (total number of edges - remaining edges in RG_Del_Eges)</td>
</tr>
<tr>
<td>Number of Edges Predicted using Random</td>
<td>90</td>
</tr>
<tr>
<td>Number of Edges Predicted using Wordnet-Cosine similarity</td>
<td>117</td>
</tr>
</tbody>
</table>

Table 6.14: Comparison between Random Accuracy and Wordnet-Cosine distance Accuracy when Distance Threshold = 60 and for different values of % edges deleted

<table>
<thead>
<tr>
<th>% edges del</th>
<th>Rand Acy</th>
<th>dist Acy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>43.82</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>53.37</td>
<td>100</td>
</tr>
<tr>
<td>0.3</td>
<td>51.49</td>
<td>100</td>
</tr>
<tr>
<td>0.4</td>
<td>48.17</td>
<td>100</td>
</tr>
<tr>
<td>0.5</td>
<td>49.32</td>
<td>100</td>
</tr>
<tr>
<td>0.6</td>
<td>47.1</td>
<td>100</td>
</tr>
<tr>
<td>0.7</td>
<td>52.88</td>
<td>100</td>
</tr>
<tr>
<td>0.8</td>
<td>51.26</td>
<td>100</td>
</tr>
<tr>
<td>0.9</td>
<td>48.94</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 6.15: Difference between the modified Wordnet-Cosine distance and the Random Graph adjacency matrix

<table>
<thead>
<tr>
<th>alpha (Distance Threshold)</th>
<th># Different Elements</th>
<th>%of difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>131298</td>
<td>0.192158412</td>
</tr>
<tr>
<td>9</td>
<td>131604</td>
<td>0.192606252</td>
</tr>
<tr>
<td>8</td>
<td>132082</td>
<td>0.193305819</td>
</tr>
<tr>
<td>7</td>
<td>132842</td>
<td>0.194418101</td>
</tr>
<tr>
<td>6</td>
<td>134260</td>
<td>0.196493385</td>
</tr>
<tr>
<td>5</td>
<td>136704</td>
<td>0.200070249</td>
</tr>
<tr>
<td>4</td>
<td>141334</td>
<td>0.206846388</td>
</tr>
<tr>
<td>3</td>
<td>151238</td>
<td>0.221341178</td>
</tr>
<tr>
<td>2</td>
<td>173796</td>
<td>0.254355462</td>
</tr>
<tr>
<td>1</td>
<td>234052</td>
<td>0.342541857</td>
</tr>
</tbody>
</table>
Chapter 7

Conclusion

Finding similarity between profiles in a complex/social network continues to be an issue of great interest to the network research community. The main approaches to this problem can be divided into two classes: structural/topological approaches and content/semantic approaches. This thesis takes as inspiration work in both of these approaches to produce a unified similarity measure, Wordnet-Cosine, based on a structural approach applied to semantic encoding of profiles. As its name suggests, Wordnet-Cosine similarity uses Wordnet, a lexical database, to extract and represent the semantic content of a profile, and combines this with Cosine-similarity, well known from research in information retrieval.

To illustrate the behavior of Wordnet-Cosine similarity, several experiments have been conducted on three data sets: the Facebook (and extended Facebook), the DBLP, and the Movielens. The results for the Wordnet-Cosine similarity have shown a relatively more significant performance over the other similarity measures when applied on one dataset, Movielens, and comparable performance for the other two data sets. When used to predict links the Wordnet-Cosine similarity outperformed the random method in predicting deleted links from a random social network.

Future Work

The development of, and experiments with Wordnet-Cosine reported in this thesis have opened up potential new directions in the research of semantic and structural similarity measures for complex networks. These include (but are not limited to) (1) expanding the second experiment (link prediction) by using other similarity measure, (2) combine explicitly an edge similarity measure
into Wordnet-Cosine similarity. (3) Integrate the movies genres information obtained from IMDB with Facebook dataset to possibly evaluate the similarity measures. (4) Experiment with larger datasets to evaluate the scalability of different similarity measures.
Chapter 8

Appendix

8.1 Code

8.1.1 Clustering Code

```matlab
% clustering and silhouette
% replaced the D_Cat_FD_N_F by SM_Cat_FD
opts = statset('Display','final');
idx = KernelKmeans(D_Cat_FD_N_F,6);

plot(D_Cat_FD_N_F(idx==1,1),D_Cat_FD_N_F(idx==1,2), 'rx','MarkerSize',12);
hold on
plot(D_Cat_FD_N_F(idx==2,1),D_Cat_FD_N_F(idx==2,2), 'bo','MarkerSize',12);
hold on
plot(D_Cat_FD_N_F(idx==3,1),D_Cat_FD_N_F(idx==3,2), 'go','MarkerSize',12);
hold on
plot(D_Cat_FD_N_F(idx==4,1),D_Cat_FD_N_F(idx==4,2), 'mo','MarkerSize',12);
hold on
plot(D_Cat_FD_N_F(idx==5,1),D_Cat_FD_N_F(idx==5,2), 'yo','MarkerSize',12);
hold on
plot(D_Cat_FD_N_F(idx==6,1),D_Cat_FD_N_F(idx==6,2), 'co','MarkerSize',12);
hold on
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster 6','Location','NE');
hold off
```

Figure 8.1: Clustering the D_Cat_FD_N_F matrix (the Semantic Category distance matrix)

8.1.2 Avg Silhouette Code

Figure 8.2 is the Matlab code for finding the Average Silhouette. It can be noticed from the Figure that it uses the KernelKmeans algorithm which is explained in the next section.
8.1.3 Modified Kmeans (KernelKmean) Code

Figure 8.3 is the modified Kmean Code, KernelKmean which was used to cluster the distance matrices of each measure.

8.1.4 Link Prediction Experiment

Figures 8.12, 8.5, 8.6, 8.7, 8.8, 8.9, 8.10 and 8.11 are the Matlab codes for the link prediction Experiment.

8.2 System specification

All experiments were carried out on a laptop that has the specifications detailed in Figure 5.2 and Table 8.1

<table>
<thead>
<tr>
<th>Table 8.1: System Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
</tr>
<tr>
<td>Memory (RAM)</td>
</tr>
</tbody>
</table>

Table 8.2 shows the software used for experiments conducted through the thesis work:
Table 8.2: Used software

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matlab</td>
<td>8.1.0.604 (R2013a)</td>
<td>For clustering and silhouette, histograms, spearman correlation</td>
</tr>
<tr>
<td>Java Eclipse</td>
<td>4.2.0</td>
<td>Compute the similarity measures using different datasets, find statistics about Facebook dataset</td>
</tr>
<tr>
<td>Excel</td>
<td>2010</td>
<td>Histograms, scatterplot, correlation</td>
</tr>
<tr>
<td>Visual Studio .Net</td>
<td>2010 For downloading the Facebook dataset</td>
<td>2013 used to download the friends of the fb users listed in Skullsecurity list, which was later extended to include the friends movies interest</td>
</tr>
</tbody>
</table>
Figure 8.3: KernelKmean: the modified Kmeans which takes a distance matrix as an input
Figure 8.4: Link Prediction Experiment - part 1
Figure 8.5: Link Prediction Experiment - part 2
Figure 8.6: Link Prediction Experiment - part 3
Figure 8.7: Link Prediction Experiment - part 4 (a)
Figure 8.8: Link Prediction Experiment - part 4 (b)
Figure 8.9: Link Prediction Experiment - part 4 (c)

```matlab
% %) LINK PREDICTION - RANDOM
% Predict edges randomly and experiment with varying the number of inserted edges.
% or insert 'EdgInserted_Sim' edges at random positions in the RG_Del_Edges matrix
% or insert at the same positions were we deleted edges.
PredictedLinks_Ran = 0;
N_Randoms = N_EdgesToDelete; % insert N edges equal to the number of deleted edges.
x = round(rand(N_Randoms)); % generates N Randoms boolean numbers (1, 0)
x_CopyN = 1; % an idnex used to retrieve binary numbers (1,0) which will be used to predict links if the number retrieved is 1.
for i = 1:size(RG_Del_Edges_Ran,1)
    for j = 1:size(RG_Del_Edges_Ran,2)
        if (RG_Del_Edges_Ran(i,j) == 2 & & RG(i,j) == 1)
            if (x_CopyN < length(x))
                if (x(i) == 1) % predict this link if x(i) == 1, i.e. predict links based on random binary (0,1) numbers.
                    PredictedLinks_Ran = PredictedLinks_Ran + 1;
                    RG_Del_Edges_Ran(i,j) = 1; % Predicted link
                end
            end
        end
    end
end
end
end
% %) EVALUATION
% Calculate a matrix that includes the positions of the predicted edges.
% Find the difference matrix between the prediction of missing links using
% the Wordnet-Cosine Distance and the random method.
PredictedLinks_Sim_M = xor(RG_Del_Edges_Sim, RG_Del_Edges);
PredictedLinks_Ran_M = xor(RG_Del_Edges_Ran, RG_Del_Edges);
```

Figure 8.10: Link Prediction Experiment - part 4 (d)
Figure 8.11: Link Prediction Experiment - part 4 (e)
<table>
<thead>
<tr>
<th>View basic information about your computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows edition</td>
</tr>
<tr>
<td>Windows 7 Home Premium</td>
</tr>
<tr>
<td>Copyright © 2009 Microsoft Corporation. All rights reserved.</td>
</tr>
<tr>
<td>Service Pack 1</td>
</tr>
<tr>
<td>Get more features with a new edition of Windows 7</td>
</tr>
</tbody>
</table>

System

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Acer</td>
</tr>
<tr>
<td>Model</td>
<td>Aspire 5750G</td>
</tr>
<tr>
<td>Rating</td>
<td><img src="acer.png" alt="Windows Experience Index 5.4" /></td>
</tr>
<tr>
<td>Processor</td>
<td>Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz 2.40 GHz</td>
</tr>
<tr>
<td>Installed memory (RAM):</td>
<td>6.00 GB (5.85 GB usable)</td>
</tr>
<tr>
<td>System type</td>
<td>64-bit Operating System</td>
</tr>
<tr>
<td>Pen and Touch:</td>
<td>No Pen or Touch Input is available for this Display</td>
</tr>
</tbody>
</table>

Figure 8.12: System Specifications
Bibliography


[33] Cuiping Li, Jiawei Han, Guoming He, Xin Jin, Yizhou Sun, Yintao Yu, and Tianyi Wu. Fast computation of simrank for static and dynamic information networks. In Proceedings of the 13th International Conference on Extending Database Technology, pages 465–476. ACM, 2010.


