Masterarbeit

Titel der Masterarbeit

Social Network Analysis in DBpedia

Verfasser

Miki Alvin Zehetner, Bakk.

angestrebter akademischer Grad

Master of Science (MSc.)

Wien, im Oktober 2010

Studienkennzahl lt. Studienblatt:	A 066 950
Studienrichtung lt. Studienblatt:	Masterstudium Informatikdidaktik
Betreuer:	Univ.-Prof. Dr. Wolfgang Klas
Mitbetreuer:	Dr. Bernhard Haslhofer

Abstract

Daily Life is more and more affected by modern forms of communication and media. In the world of today, we live our lives within network based environments. We check e-mails, make mobile phone calls and interact on social media platforms – starting from Facebook or Twitter up to Wikipedia. The high volume of raw computable data leads to research topics about social network analysis. Using this method, it is possible to reveal distinct patterns of interaction. Depending on the communication media, it allows the investigation of behavioral patterns of strong and weak relationships, relationships of liking and disliking someone, or even dividing important actors from less-important actors within a network system.

Besides, network technology does not stand still. It is constantly expanding, enhancing and restructuring itself. A great new vision of the World Wide Web is the enhancement to uniform standards on the underlying data to a Web of Data. The Web of Data, or Linked Data, already has a huge community and a fast growing amount of freely accessible, machine-readable data. The nucleus and crystallization point of the Web of Data is DBpedia, which provides a machine-readable representation of the entire Wikipedia contents as Linked Data on the Web.

This thesis seeks to connect the data of Linked Data with the method of the social network analysis. In order to achieve this, we would like to extract networks from DBpedia and analyze the extracted actors to draw a valid conclusion about using DBpedia as a source of data for social network analysis. To assure that social network analysis on DBpedia is possible and reasonable, we will exemplarily analyze networks of writers, scientists, soccer players and architects to answer questions like “Who is the most important writer/scientist in history?”, “Which transfer patterns do soccer players follow?” or “Do architects work in teams?”. Another topic of this thesis is the usability and usefulness of this whole approach in social science.

Zusammenfassung

Unser Leben verlagert sich immer mehr in Richtung netzbasierter Umgebungen. Wir schreiben E-Mails, telefonieren mit Mobiltelefonen und kommunizieren mit Freunden in Social Media Plattformen, von Facebook bis Wikipedia. Das schafft eine große Anzahl an verwertbaren Daten für die Soziale Netzwerkanalyse. Diese Methode erlaubt es, basierend auf dem Medium spezielle Kommunikations-Schemata zu analysieren, Verhaltensmuster bei starken und schwachen Beziehungen, Beziehungen bei denen sich Akteure mögen oder ablehnen zu untersuchen. Mit ihr kann man auch Aussagen treffen, wie wichtig einzelne Akteure in Relation zu anderen im Netzwerk sind.

Netzwerk Technologien entwickeln sich kontinuierlich weiter. Ein gutes Beispiel dafür ist die Erweiterung des World Wide Web zum sogenannten Web of Data. Hier werden Standards geschaffen, um die den Webseiten zugrunde liegenden Daten einheitlich, offen und maschinenlesbar zu gestalten. Das Web of Data, auch Linked Data genannt, hat eine große Gemeinde und eine schnell wachsende Anzahl an frei verfügbaren, maschinenlesbaren Daten. Das leuchtende Zentrum dieser verlinkten Daten ist die DBpedia, welche Daten aus der Wikipedia extrahiert und anhand der Linked Data Prinzipien aufbereitet.

Diese Arbeit versucht die frei verfügbaren Daten des Web of Data mit der Methode der Sozialen Netzwerkanalyse zu verbinden. Um das umzusetzen, wollen wir Daten von der DBpedia extrahieren und die extrahierten Akteure analysieren, um daraus konkrete Aussagen herleiten zu können. Konkret möchten wir jeweils ein Netzwerk von Schriftstellern, Wissenschaftlern, Fußballspielern und Architekten extrahieren um, unter anderem, Fragen zu beantworten wie „Wer ist der wichtigste Schriftsteller/Wissenschaftler der Geschichte?“, „Welchen Transfermustern folgen Fußballspieler?“ und „Arbeiten Architekten in Teams?“.

Die Beantwortung solcher Fragen gibt Aufschluss darüber, ob die Soziale Netzwerkanalyse in Verbindung mit der DBpedia grundsätzlich möglich ist. Auch Ziel dieser Studie ist es, herauszufinden ob dieser Ansatz brauchbar ist für die Sozialwissenschaft.

Content

Part I – Background and Related Work

Part 2 – Methodoloigy, Implementation and Proof of Concept

List of Figures

List of Tables

List of Listings

1 Introduction

Social network analysis represents an interdisciplinary research area, where sociologists and computer scientists bring their competence together. Sociologists possess particular knowledge in the accurate editing and correct interpreting of empirical data in their research fields. Computer scientists have, amongst others, the knowledge for parsing and processing the needed data from the web.

Social network analysis is used within communication networks, especially in economical contexts, and relationship networks. Communication networks depend on e-mail data, chat histories, mobile phone call records, etc.. Relationship networks can either be simple ones like networks of “who knows whom”, or friendship networks. More complex relationship networks have different relationship attributes, such as networks of liking and disliking, strong or weak relationship networks, etc..

Within the social web, human-to-human communication is shifting more and more towards online communication possibilities. Online communication data is usually recorded and can be computed to big-scale social networks. Social software, such as Facebook, Orkut, StudiVZ or Frienster, provides an even greater dimension of creating social networks.

However, social networks do not have to be about living people in the first place. There is also the possibility to use this method with historical data. Data on the Roman Empire, for example; every year two consuls were elected, who served and worked in an administration of the highest political office. We can say, every consul is connected to the other consul he served with. This assumption is the base of a social network.

In this thesis, we want to apply social network analysis methods on Linked Data. Linked Data is part of Tim Berners-Lee's Semantic Web vision. It unites the underlying data of websites, makes them accessible, interlinks them and (most importantly for this thesis) exposes information in a machine-readable way. Linked Data has a growing research community and an increasing number of data sources.

One of the biggest and most famous Linked Data sets is the DBpedia [67,68]. This project processes data from Wikipedia and transforms it according to the Linked Data principles. This leads to an ample amount of informational data.

Until now, social network analysis in the area of Linked Data is only applied on personal profiles expressed in the FOAF vocabulary, which is the underlying vocabulary of social software data. This thesis will investigate the possibilities of using the method of social network analysis on other Linked Data sources. As a proof of concept we will specially focus on analyzing social networks extracted from DBpedia.

1.1 Problem Description

The aim of this thesis is to investigate whether or not data from DBpedia is usable and useful for the social network analysis. In order to do so, we need to show that it is possible to properly extract and analyze social networks. Thus, we need a framework and an application that uses the interfaces from Linked Data sets. It should convert the data to a network and then compute and present the analysis results.

As a proof of concept we exemplarily extract four networks: A network of influential writers with the aim of investigating the most famous and important writers in history. Another network depends on soccer players to investigate their transfer patterns. We want to answer the question, if soccer player transfers are regional or international. The third network refers to historical scientists. There we will find out, who is, retrospectively, the most important scientist. The fourth network concerns a network of architects to investigate which buildings architects work on in teams. The network of soccer players and the network of architects are especially suited for finding errors in the extraction data and their heritage.

Additionally, to make sure this approach is not only useable but also useful for social research, a usability study will be conducted to theoretically assure or disprove whether it is eligible for social research or not.

1.2 Contribution

The contribution of this thesis is the expansion of social network analysis on arbitrary Linked Data sources. Furthermore, the thesis should find out, whether DBpedia data is interesting for social research according to social network analysis. We will provide a framework for extracting, converting and analyzing data from Linked Data sources. Data extraction capabilities are on single RDF sources, which could be one single file or a single SPARQL endpoint. The data will be converted to an internal graph structure. To be compatible with other software, the graph is convertible to a specific format. Furthermore, there are also proper analysis algorithms integrated within the framework.

1.3 Organization

The thesis is structured into two major parts and seven chapters. The first part covers the theoretical background and related work. Here aspects on social networks will be introduced, with respect to their practical use for observing communication and relationship patterns of human beings, their classification into complete, partial and ego-centric networks, as well as a common extraction methods (chapter 2). Thereafter, the topic of the graph theory and its methods will be explored more detailed and statistical metrics for social network analysis will be introduced (chapter 3). Finally, within this part we introduce the second main issue of this thesis – the topic of Linked Data. Here we present the underlying concepts and working technologies. Furthermore, we take a closer look at DBpedia and present extraction possibilities on Linked Data sources.

The second part deals with Methodology, Implementation and the Proof of Concept. Within this part, we present the methodology of our approach, divided into the approach for our extraction and analysis tool, the network analysis procedure, and the approach for the usability study (chapter 5). Afterwards we present the implementation of the SocioCatcher framework and application (chapter 6). Finally, we analyze the extracted networks and present the results of the usability study (chapter 7).

Part 1

Background and Related Work

2 Social Networks

This chapter gives an introduction to social networks, their ancestry, definition, categorization and research topics. It also illustrates a very common extraction method. In this thesis, the term social network is used for graphs with special properties to investigate living beings and therefore it is mainly a topic of social science.

2.1 Origin

Social science investigates the aspects of human society. According to this, social science is an ample field with very interdisciplinary characteristics. Amongst others, the main branches of social science are sociology, communication studies, education, political science, economics, anthropology, geography, history, law and linguistics. There are, however, also other scientific research areas of social science. For instance, social science is also integrated within health care, environment, work, arts, education, demography, culture, economy, commerce, sports, police, traffic, urbanity, (governmental and non-governmental) organizations, youth culture, globalization, technology and even within scholarship itself [63].

The quantitative methodology of social science generally consists of creating, surveying, and analyzing questionnaires. The most commonly used qualitative methods are observations, field experiments, artifact or text analysis and oral or written inquiries (e.g. interviews).

All methods take examples from an investigated group. This means, we have, for instance, a group of 8 million people, which is about the population of Austria, and, according to Alemann [4], take a random sample of 2.000, in consideration of an equal amount of all demographic groups. The social network analysis is the only method in social science which has the nature to investigate data of complete groups [34].

2.2 Definition

A network has two properties, vertices (or nodes) and edges (or arcs/links). Vertices are visualized as nodes with edges in between. Figure 1 is a simple directed network of vertices and edges.

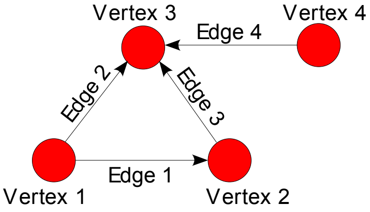


Figure 1: Vertices and edges of a directed network

The vertices of social networks are called actors and can be of the following types [2]:

Human beings	Animals
Small groups	Economic organisations
Social classes	Occupations
Nations	World alliances

Table 1: Vertices of social networks

The edges are called relationships and can be signed or unsigned. Unsigned networks are simple relationship or communication networks. Signed networks give models a more realistic behavior [9]. Signed edges can be weak or strong ties, like or dislike, or even more capabilities for more complex networks.

Social network analysis focuses on patterns of actors and their relationships. It is used for many different kind of networks, like communication networks, friendship networks, enterprise networks, health networks, networks of innovation, etc. [3].

2.3 Social Network Research

Social networks are part of the social science since its beginning in the middle of the 19th century. The famous philosopher Karl Marx already wrote in 1857: "Society is not merely an aggregate of individuals; it is the sum of the relations in which these individuals stand to one another." [1]. These Social network metaphors were used intuitively over a long time [2]. Social networks are, in most cases, very ample (many actors with many connections), so over the last few centuries they were too complex to visualize and compute.

Technical possibilities changed and a new computational social science appeared [5] with its flagship, the social network analysis. With the advent of social media, social software and the social web, a huge amount of recorded material is now appearing, which can be used for analyzing social behavior. There are e-mails, instant messengers, message boards and many other communication networks. Besides this, friendship networks become more and more part of the daily routine for millions of people. With these networks it is possible to communicate in many different ways, play multiplayer games, organize events and invite friends, etc.

But that's not everything. Research is also being conducted into so-called “sociometers”, which are electronic devices that can be worn by people to record movement, location, proximity, and other measures [6]. The data can be used to obtain knowledge about face-to-face group interaction and group-dynamics, i.e. in companies, but also to analyze how diseases can spread. The proximity and time data of mobile phone calls are also collected and analyzed to get patterns on social communication behavior [8].

There is also research in simulating complex macro social networks to get a better overview on society and how society is changing over time [7]. There are different ways to reveal network complexity on large networks. One is to group nodes, another is to analyze grouped edges, so called “link communities” [10].

Phone companies collected records of phone calls made by their customers over many years. Social software providers have been recording data of chat and other interactions over a long period of time.

Computational social science analyzes this kind of data in order to draw conclusions based upon this. This leads to problems, for instance, if Google Research or Yahoo Research work on their data, the knowledge they produce can not be reproduced by anyone within the scientific community, for fear of breaching privacy regulations. For example, the Facebook API allows programs to fetch friends of a Facebook user, but the policy does not allow it to store the data. It is the hardest challenge to respect access and privacy on the one side, but to still get as much data as possible, because only a broad amount of raw data can lead to a profound knowledge.

2.4 Communication Networks

With social research, communication patterns are often used for analysis. There is a great amount of data available like phone call lists, e-mail lists from companies, blogs and webpages, instant messenger histories, social software records or even special devices and mobile phone lists for spatial data.

To gain a better insight into online communication capabilities, Figure 2 lists a categorization for different kinds of communication types and their software in working or private environments.

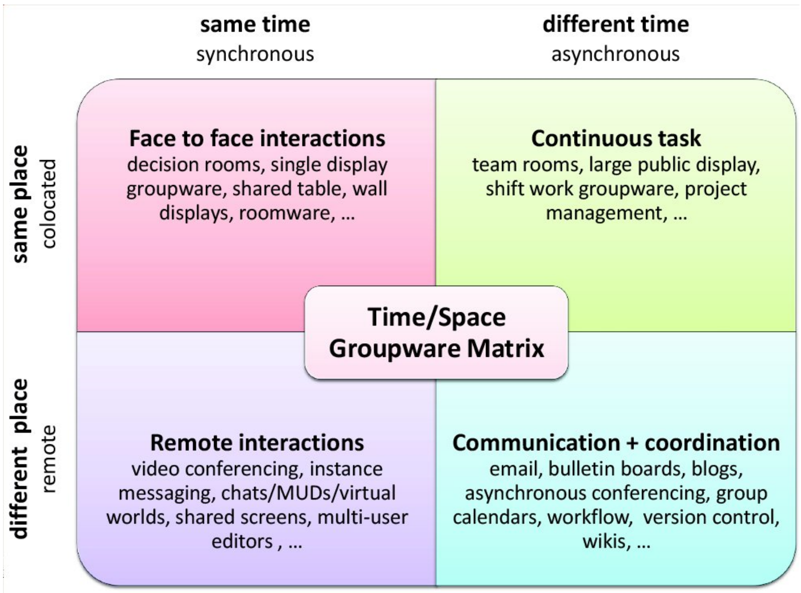

same time
synchronous
different time
asynchronous
same place
colocated
different place
remote
Face to face interactions
decision rooms, single display groupware, shared table, wall displays, roomware, …
Continuous task
team rooms, large public display, shift work groupware, project management, …
Time/Space Groupware Matrix
Remote interactions
video conferencing, instance messaging, chats/MUDs/virtual worlds, shared screens, multi-user editors , …
Communication + coordination
email, bulletin boards, blogs, asynchronous conferencing, group calendars, workflow, version control, wikis, …

Figure 2: Computer Supported Cooperative Work Matrix (CSCW)
Ressource: http://upload.wikimedia.org/wikipedia/commons/2/28/Cscwmatrix.jpg

The CSCW matrix shows different communication capabilities. Communication patterns on proximity communication (the first column) is commonly used in economic environments. “Different place” communication has a bigger focus in science, because this data can be easier recorded and computed for big scale networks.

2.4.1 Phone Call Networks

Mobile phone call lists or those of landline phones are often used for analyzing highly complex social networks but also for enterprise communication networks for optimizing workflows. This is often motivated by economical reasons, because communication is a very important way to optimize productivity and costs. First, there has to be knowledge about communication structure (phones, e-mail, chats, ...) in a company, followed by making it more effective, for instance, by training or even by finding employees who don't participate in the company structure and exchange them as a last resort.

Understanding phone call networks is also a broad section in research. Mobile phone calls especially are of great interest to the scientific community, which produced a large amount of literature on mobile phone networks [8,10-16]. Thus mobile phone calls are not the only possibility for social agents to interact, there is an effort to understand dynamics and behavior patterns on mobile phone users, because of the fact that nearly everyone owns at least one mobile phone. This leads to complete data sets. Mobile phone data has a potential value for public traffic engineers, safety managers, emergency response personnel, city planning and resource management. This data gives insight into what humans do in their daily routine, into group dynamics within crowds as well as how individuals change their behavior in emergencies like traffic jams, protests or riots [14,15]. For more details see Section 2.6.

A scientific study has conducted research into the persistence and dynamics of communication ties between actors [11]. It shows that, among others, reciprocal "two-way" links are the most persistent. Another study analyzed the changes of the whole network over time and concluded that if complex communities, to be persistent, have to embed new actors very fast, which means the network need to be very dynamic. Small communities have to be static, for persistence. Figure 3 illustrates a time stamp of a phone call network [13].

Even the largest social network in 2007 (3.9 million nodes) is a network of mobile phone calls [16]. This network displayed interaction strength and cliques, or communities, arising from it and found a global manifestation of the weak-ties hypothesis.

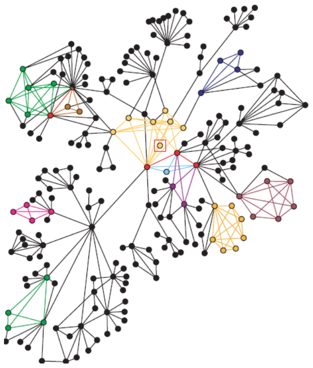


Figure 3: Phone call network [13]

2.4.2 Internet Networks

Online social networks are networks extracted from e-mail logs, blogs and other web pages or social software. Social networks from e-mail logs were first captured to demonstrate the differences between the command structure and the communication structure in organizations [17], differences in online and offline communication [18], as well as to explain e-mail overload [19]. But research on e-mails is not as easy as it appears at a first glance. E-mails have many functions. They are for sending information, sharing files, act as a contact manager or a mass mailing outlet. All these functions are technically not distinguishable and therefore hard to extract properly.

There are two approaches for capturing e-mail data, server-side and client-side. With server-side captures, it is possible to get the network for a whole domain, such as in companies. Client-side captures are well suited for personal networks, to compare among themselves. Normally these networks are directed and weighted. To avoid privacy issues, e-mail bodies should be cut off.

Another type of communication network can be extracted via web pages like blogs. The web is naturally a hypertext network. In assumption that web pages stand for the opinions of certain people, social networks can be captured. In 2004 a social network of political blogs was built to show the spread of personal opinions. Figure 4 shows the community-separation between republicans and democrats [20]. This kind of data is collected by scrapers and spiders via hypertext analysis.

Social software like Facebook, LinkedIn, Wikipedia or Friendster can also be used to extract communication social networks. A research on Facebook analyzed 4.2 million anonymized nodes of college students and their 362 million messages and "pokes" during a 26 month interval. The data was observed on annual routine of the users as well as their communication and social lifes including seasonal variations [21]. Another work analyzed communication networks to give more information on personal relationship of a Facebook "friend". Especially reciprocal communication gives information of a friendship connection. On Figure 5 it is obvious, a network of all friends is not very valid for a persons friendship behavior. A better, informative fact for this is, that Facebook users have up to 10 "friends" with a reciprocal communication [22].

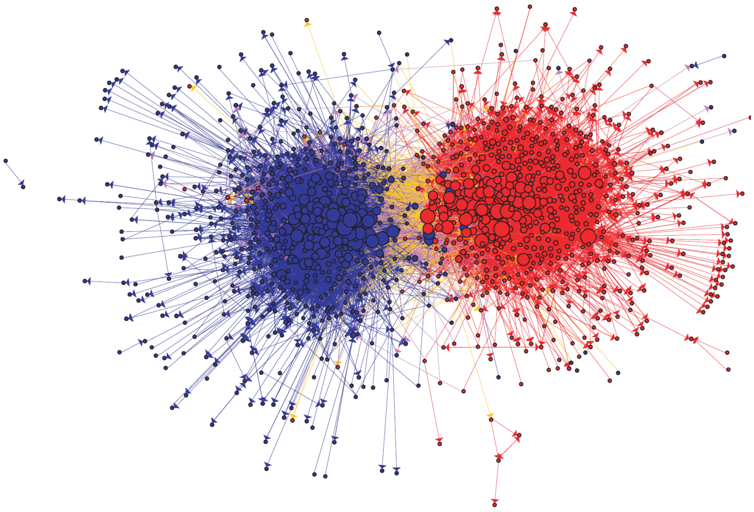


Figure 4: Social network of political blogs

The blue nodes illustrate democrat -, red ones republican blogs [20].

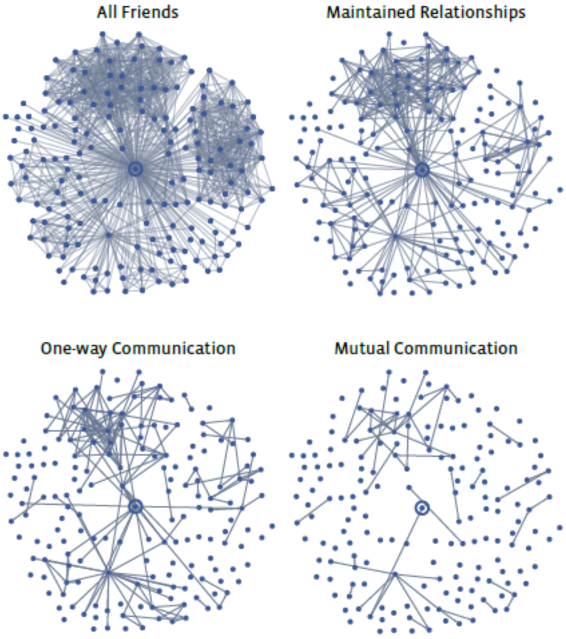


Figure 5: Ego-centric network of a facebook user [22]

2.5 Relationship Networks

In the most cases, relationship networks are more intuitive than communication networks. Communication is strictly measurable, but friendship is not that easily measurable and scalable. How do we categorize the relationship between two persons? Simple relationship networks are family trees or corporate networks (hierarchical enterprise networks, network of a corporative state, such as the traditional Indian caste system), where a relationship is predefined. If there is no predefined definition, the correct extraction of data has to be done carefully.

Regarding the Facebook example mentioned before (see Figure 5), where we have a network of friends, the question arises: What does “friends” mean in the context of Facebook? Some people collect Facebook-friends like money, the more the better. A network of Facebook friends does not have a reference to a person's social life, and his or her real-time friends. One indicator to use, as mentioned above, is reciprocal communication of two users to guess a real relationship [22].

2.5.1 Social Software Networks

In social software systems, such as Facebook, everyone has the possibility to create his own network of friends. Several social networking platforms publish anonymized data periodically. Nevertheless, advanced analysis is only possible for research labs of the own company.

Most works in social network analysis focus on positive relationships between two actors. Possible meanings of edges are “has a professional relationship”, “are friends” or “know each other”. But these edges are not very realistic for elaborate social behavior. Modern social network analysis focus on complex weighted links.

A very famous link weighting is weak and strong ties [23]. This splits social behavior in friends and acquaintances. The renowned theory by Granovetter says, that weak ties are more important for getting a job than strong ties. Another link weighting is the proof of the theory of social balance from 1958 by Heider (Figure 6) for large-scale networks [24]. Beside that, like and dislike triad networks and their temporal evolution are focused on current social network researches [25].

my friend’s friend is my friend
my friend’s enemy is my enemy
my enemy’s friend is my enemy
my enemy’s enemy is my friend

Figure 6: Heider's social balance theory

Used data sources are the trust network of Epinions, where people link to other users indicating trust or distrust. This data set is used for instance by eBay. Special blogs, such as Slashdot, are used, where users can choose other users as “friends” or “foes”. Other social software that uses “yes or no” polls, like Wikipedia, is also a good source for extracting like and dislike networks. Wikipedia uses such polls when administrators are chosen and every user can decide if he pleads for or against him.

2.5.2 FOAF Networks

Friend of a friend (FOAF) networks consist of RDF data and give the possibility to interlink different online social networks together. FOAF is one of the most used semantic web ontologies with nearly a million *foaf:person* attributes [28]. Although, this ontology is not fully established on the web, the consisting data is becoming more and more popular in social network research [27]. FOAF data sets can easily be searched by Swoogle, a crawler-based indexing retrieval engine for the semantic web [29]. For example, Flink, an ample online social network extracting tool, uses FOAF as one source for their social network, next to web and publication mining from Google and Google scholar and e-mail lists [30].

Since 2008 there is a new way to get data from FOAF networks very easily by using the Google Social Graph API [31]. This API uses public Facebook data and other social software and merges them to a new network with “me” and “friend” edges. So, for example, a Facebook account is connected through a “me” link to the MySpace account of the same person and each of the two nodes has his/her friends. In 2010, Facebook launched its own graph API with more specific features. Within this API there are events, groups, links, notes, photos, videos and users among other things as objects available [64].

In this thesis, FOAF will be shown as being not the only semantic web ontology to capture social networks.

2.5.3 Other Relationship Networks

As mentioned above, there is a use of historical data. An example of historical data captured from plain HTML is a project, in which every game and every player from the Dutch soccer team has been captured, beginning with Holland vs. Belgium in 1905 and, under the assumption that every player playing in the same game is related with the other players, this has been transformed to a social network [26].

Another famous approach in social network research is the creation of a co-authorship network [13]. Compared to a communication network, a co-authorship network is much more clustered. Co-authorship networks are extracted using online publication libraries or publication search engines like Google scholar. The collected data is transformed to an undirected, weighted social graph. For an example see Figure 7.

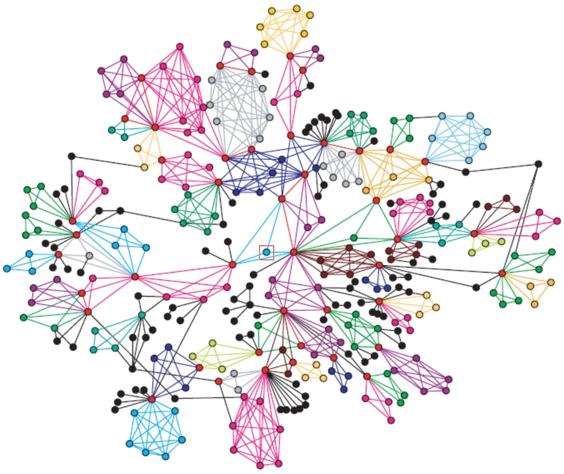


Figure 7: Co-authorship network [13]

2.6 Unusual Networks

There is a category of networks that are not social networks as described in Section 2.2, but are still usable networks for analyzing social behavior. One important type of network to analyze group behavior is a spatial network of movement. As described in Section 2.4.1 spatial data has a potential value for public traffic engineers, safety managers, emergency response personnel, city planning and resource managements. This data gives an insight into what humans do on a daily basis, group dynamics within crowds and how individuals change their behavior in emergencies like traffic jams, protests or riots [14,15].

For instance, mobile phone data can be mapped in realtime to understand, prevent and avoid traffic jams. Nonetheless such innovations are greatly opposed to such "big brother" projects [31]. Another social study collected anonymized spatial data on 100,000 mobile phone users to analyze reproduceable travel patterns. This had the aim of gathering information on human movement within the daily routine of people [12]. As shown in Figure 8, most individuals travel only short distances, but some are also traveling many kilometers.

Other scientists are working on a sensor package for mobile phones to collect more relevant data than would be collected from merely the position of a human being. For instance, it collects pollution values [32].

An approach in such mobility patterns is the investigation into new computer viruses that spread via limited communication protocols (such as WLAN, Bluetooth, ...). The spread of these viruses depend on the proximity of two mobile phones, which closes the analogy to biological viruses [33].

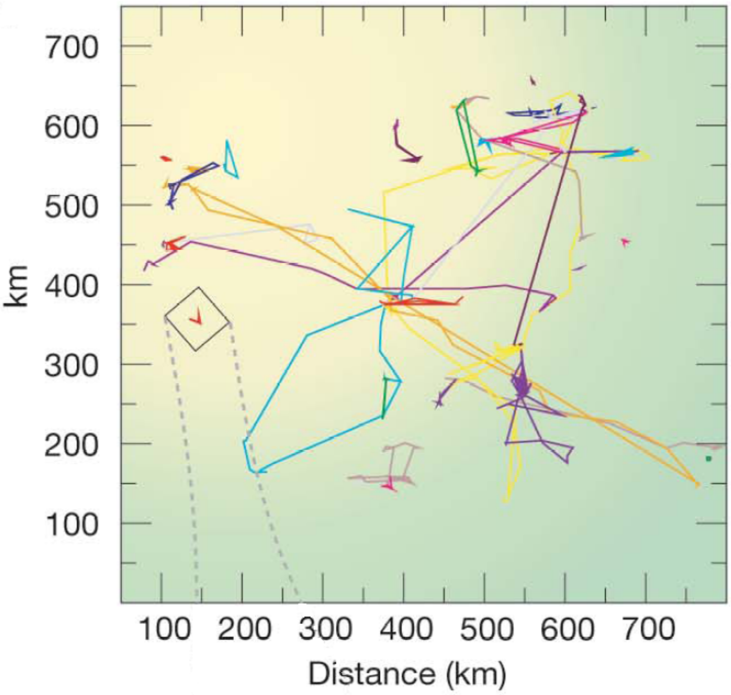


Figure 8: A week-long trajectory of 40 mobile phone users [12]

2.7 Network Extraction

So far, the existing networks have been categorized from a user's perspective. Now it is possible to take a closer look at the network creation and its main questions:

What kind of behavior will be observed? What underlying data is available? Is the data applicable or does it need to be computed in a more complicated way? Are the resources readily available to extract complete networks and would this process be appropriate for this concept? If not, are there any other opinions like partial or ego-centric networks being useful? This is only a few of many major questions that everyone who extracts a social network has to deal with, but very essential ones nevertheless.

In the following, three types of networks will be presented and methods for transforming data to social networks described.

2.7.1 Complete Networks

Complete social networks are surveyed by observation, statistical data pooling, questioning or other methods as mentioned already in this chapter. Who belongs to the network and who does not? The researcher has to select his network-players carefully. For instance, a scientist can take the participants of his course for a social network, or he can take scientists that published in the past 5 years in the 5 most renowned scientific journals. Figure 4 on page 11 shows such a a deliberate selection of blogs. The creator of this network chose consciously only political blogs and renounced non-political ones.

In an empirical survey, it is not only necessary to think about who the are actors are, but also how they are related. Do we wish to know who is (binary) related to whom, or do we even want to know how strongly they are related? Is the relationship one-way or reciprocal?

Most networks in social network analysis are complete ones.

2.7.2 Partial Networks

Partial networks are often used, if there are too many people to survey. For instance, if we wanted to make a network of all people of India, it would take a lot of work. Therefore it would be more efficient to take a selection of people. A commonly used principle is the pyramid scheme, where the first level of participants are asked “Who can you recommend for questioning?”. The resulted new participants are the second level participants, and so on. This kind of network is not used very often, because the chances of getting a specific demographic group rather than a representative example of the targeted group are very high. With a lack of examples on partial networks, Figure 8 on page 14 is indeed no usual social network, but it shows a selection of 40 mobile phone users and can be viewed as a partial network.

2.7.3 Personal Networks

Personal, or ego-centric, networks are a subgroup of partial networks and are commonly used when huge amounts of data needs to be processed. For this kind of network, we have to choose one actor, called *ego*, of a network and take all his friends, called *alter*, and their relations amongst each other. Then we delete ego and analyze the new network. How dense is the network? How many connected components does the network have?

An example of ego-centric networks is illustrated on Figure 5 on page 11.

2.7.4 Indirect Extraction Procedure

A very common case is to have actors and events (or containers). For instance in a scientific research study, we had Dutch soccer players (actors) and of the dutch soccer games (events) [26]. Now every player is connected to the other players he played with in a game.

In Table 2 there is an example of CEO meetings (events, e1 to e3) and CEOs (actors, a1 to a4) viewed as affiliation matrix. There is an easy formula to get a person to person matrix out of this person to event matrix: $A = M * M^T$.

CEO	e1	e2	e3
a1	1	1	1
a2	1	0	0
a3	0	1	0
a4	1	0	1

Table 2: CEO meeting attendance

If we insert the values we get the result matrix:

$$A = M * M^T = \begin{pmatrix} 1&1&1 \\ 1&0&0 \\ 0&1&0 \\ 1&0&1 \end{pmatrix} \begin{pmatrix} 1&1&0&1 \\ 1&0&1&0 \\ 1&0&0&1 \end{pmatrix} = \begin{pmatrix} 3&1&1&2 \\ 1&1&0&1 \\ 1&0&1&0 \\ 2&1&0&2 \end{pmatrix}$$

Then convert it to an affiliation matrix with c = 2:

$$\begin{pmatrix} x&0&0&1 \\ 0&x&0&0 \\ 0&0&x&0 \\ 1&0&0&x \end{pmatrix}$$

The result is called a “socio-matrix”. To get an affiliation matrix, the result matrix can be calculated by changing every value below a certain threshold c as zero, everything above c as 1 (the diagonal doesn't matter).

For huge, complex networks the matrix computation could take a lot of resources. For this, a very simple algorithm is mentioned in Listing 1.

```
events //each event has all actors as array
result //a typical empty graph class
foreach events as event
 for i = 0; i < event.count; i++
  for j = i+1; j < event.count; j++
    if !result.edgeExists(i,j)
      result.addedge(i,j,1)
    else
      result.addedge(i,j,result.edgeWeight+1)
```

Listing 1: Algorithm to get weighted person to person edges

3 Social Network Analysis

This chapter describes the characteristics and basic principles of networks and their metrics for analyzing social networks in particular.

3.1 Graph Theory

Graphs are a unified structure to model pairwise relations between objects from a designated collection. As mentioned in Section 2.2, networks are a collection of vertices (or nodes) and links (or edges, or arcs). Vertices are the agents, in social network analysis *actors* (see Table 1 on page 8), in technical networks *clients*, *hosts*, *router*, *bridges*, in HTML DOM trees nodes are different tags like *html*, *head*, *body*, *p*, *a*, *img*, etc.

Links are the relationships between vertices. Clients in technical networks are connected to a router or switches, these are connected to a provider, and so on. As mentioned earlier, social networks are divided into two groups, communication networks and relationship networks. Communication networks have their links when two actors are communicating with each other, whilst relationship networks have a certain meaning to their relation, like “has a professional relationship”, “are friends”, “know each other”, “likes/dislikes one another” or “has a strong/weak relationship”.

For a better understanding, the first paper on graph theory was used by Leonhard Euler in 1736. The issue was about finding a way around Königsberg, only being able to cross each bridge once. Euler abstracted every isle to nodes and the bridges to edges (undirected links). Euler proved, that there is no possible solution to this problem [36].

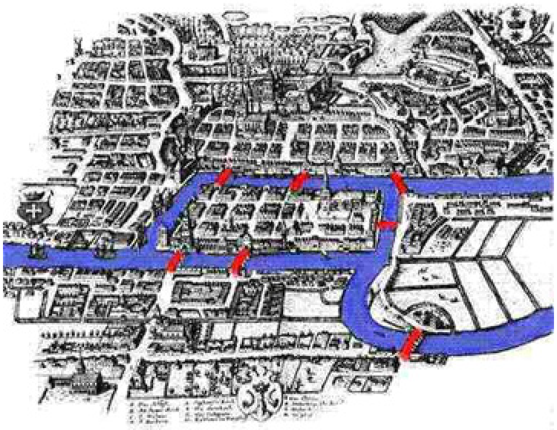


Figure 9: Königsberg bridge problem [36]

3.1.1 Undirected Graphs

Graphs can be directed or undirected. Undirected means, that there is no direction given to the graph. This is used if agents have a reciprocal relationship. These links are called edges. Undirected Networks are often used in social network analysis, but are very unusual in real life. A simple friendship network usually has no directed relations, but if a group of people (for example in school) were asked “who are your friends?” some directed relations would almost definitely be discovered. Undirected relationships are very useful in graph theory if direction does not matter. With undirected networks we can analyze how centralized actors are with respect to the whole network, or how important they are when considering data flow through the network (see Section 3.2.3). The mathematical definition of an undirected graph G is:

Graph $G=(V,E)$ consists of vertices $V\equiv(v_1,v_2,v_3,\ldots,v_N)$ and edges $E\equiv(e_1,e_2,e_3,\ldots,e_K)$, where $V \neq 0$ and every link consists of two (unordered) nodes. A node is usually referred to its order i in the set of V. Each link is defined by a couple of nodes i and j, denoted as (i, j) or e_{ij}. Undirected networks have $e_{ij} = e_{ji}$ [37].

3.1.2 Directed Graphs

Directed relationships are often denoted as arcs. Arcs give the possibility to give individual links different weighting and also to reduce them to one-way relationships. If we have a look on an asynchronous DSL connection in technical networks, the arc from the provider to the client will be associated with its download capacity and the upload capacity is attached to the arc the other way around. Another example would be a street network. Such a network has to be directed, because there could be one-way streets or streets, where we can only turn to the cross-road if we are driving in the right direction. So, directed networks give a lot more possibilities for modeling complex behavior.

In social network analysis directed networks are dominant within communication networks. For instance, an e-mail or a Facebook message is one-way. Only if a person answers to an e-mail or a Facebook message, it becomes a two-way conversation.

According to the definition of undirected networks (in Section 3.1.1), directed networks are additionally denoted by the term $e_{ij} \neq e_{ji}$.

3.1.3 Data structures

Graph data can be represented in different ways. The first, as mentioned above in the mathematical definitions, are lists. Figure 10 will be used as an example for the next listing of possible data structures.

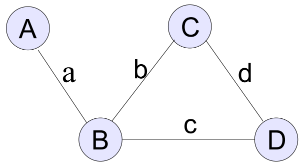


Figure 10: Undirected graph

Adjacency list

The most common list category for graphs is the adjacency list. Table 3 illustrates an example adjacency list of the graph in Figure 10. In this list, every node lists the neighbors it is directly linked with. Here the edges are not important and there is also no possibility for weighting the links. Nevertheless the space for this data structure is very small. It is also quick for many operations.

A:	B
B:	A, C, D
C:	B, D
D:	B, C

Table 3: Adjacency list

Incidence list

An adjacency list combined with an object oriented approach is called an 'incidence list'. Vertices are stored with pointers on edges according to them, instead of other vertices. In this structure, edges can be easily weighted [38].

Adjacency matrix

Another data structure for representing a graph is the adjacency matrix. This matrix is a $n\,x\,n$ matrix, where n is the number of nodes. In directed networks, the row is the outgoing node, the column is the incoming one. The diagonal in this matrix is unimportant, because it would be an edge to a node itself. In case the graph is undirected, the adjacency matrix is symmetrical, as shown in the example in Table 4. This structure is suitable for small graphs and/or performing computation by linear algebra. If we use this structure for networks with 1000+ vertices, computation can become a problem.

	A	**B**	**C**	**D**
A	0	1	0	0
B	1	0	1	1
C	0	1	0	1
D	0	1	1	0

Table 4: Adjacency matrix

Incidence matrix

The incidence matrix is a rectangular $n\,x\,m$ matrix, where the rows are indexed by vertices and the columns by edges. For directed graphs, values can be differed by +1 for an incoming edge, and -1 for an outgoing one. Table 5 gives an example for an incidence matrix.

	a	**b**	**c**	**d**
A	1	0	0	0
B	1	1	1	0
C	0	1	0	1
D	0	0	1	1

Table 5: Incidence matrix

Pajek NET format

Pajek is the Slovenian word for spider. The Pajek NET format is a very rich format with a lot of facets, which allow the representation of simple to complex networks as well as time event networks. A very simple Pajek file starts with the line “_*Vertices n_”“ where _n_ is the amount of vertices. After this line, there is the possibility to describe the vertices as mentioned in Table 6.

vertices_num	***label***	***[x,y,z]***	***[shape]***	***[changes of default parameters]***
Continuous vertex number 1,2,3..n	Label of the vertex “vertex xy”,...	Coordinates of vertex	Shape of presented vertex ellipse, box, diamond, triangle, cross, empty	Change the shape-default parameters

Table 6: Vertices description of Pajek format

Next, a line with **Arcs* for directed or **Edges* for undirected networks follows. The description of arcs and edges are described in Table 7.

v1	***v2***	***value***	***[additional parameters]***
Initial vertex number	Terminal vertex number	Link weight	Parameters for appearance of the edge

Table 7: Link description of Pajek format

In addition, there are other descriptors for links. **Matrix* followed by an appropriate adjacency matrix (with blanks and EOL in between) or *Edgeslist / *Arcslist followed by a adjacency list (also with blanks and EOL in between). Other link presentations are UCINET, GEDCOM and chemical formats.

A whole example Pajek NET file illustrating the graph of Figure 10 is listed below in Listing 2.

```
*Vertices 4
1 "A" 0.15 0.3 0.5
2 "B" 0.4  0.8 0.5
3 "C" 0.65 0.3 0.5
4 "D" 0.89 0.8 0.5
*Edges
1 2 1 l "a"
2 3 1 l "b"
2 4 1 l "c"
3 4 1 l "d"
```

Listing 2: Pajek basic example file

For further details, please refer to the Pajek manual [39].

3.2 Graph Analysis

After capturing a social network, two further steps are required to analyze a network. The first is a qualitative analysis combined with network visualization. The second is to analyze it more accurately through the use of certain metrics.

3.2.1 Network visualization

Network visualization is the common first step in network analysis. It allows the human eye to recognize obvious patterns, for example, how important specific nodes are for the whole network, or how clusters are interacting with others. In social networks nothing can inspire imagination more than an applicable mapped network or, at least, the most important parts of it. But it is just the first step and a network only interpreted by its visual appearance has no validity to serious research. Nevertheless, a visual network should not be missing within presentations or scientific papers, because it helps understanding. The focus of this thesis is not on network visualization, therefore this topic will not be elaborated upon [71].

3.2.2 Metrics for the complete Network

This section introduces network metrics for the whole network, which is the second step for network analysis.

Density

Density is the total number of edges in the network divided by the number of possible individual edges. Its value is within [0,1]. The maximum amount of arcs in a directed graph is the number of vertices multiplied by the number of vertices minus 1. The maximum amount of edges in undirected networks are additionally divided by 2. $|E|$ is the number of edges, $|V|$ is the number of vertices in the network.

$$Density_D = \frac{|E|}{|V|*(|V|-1)} \qquad Density_U = \frac{2|E|}{|V|*(|V|-1)}$$

This measure gives information about how dense or sparse a network is, compared to others. These other networks can be other existing networks or a fictional random network that becomes threshold for sparse and dense categorization.

For the undirected graph in Figure 10 the network density is:

$$Density_U = \frac{2|E|}{|V|*(|V|-1)} = \frac{2*4}{4*3} = \frac{8}{12} = \frac{2}{3} = 0.\dot{6}$$

$0,\dot{6}$ is a very high value for network density.

Connected components

Another metric is to count the connected components of a network. Connected components can be of vital important for metrics on individual nodes. Some of them need a connected network to give proper information. Algorithms for connected components in graph theory can be straight forward with breadth-first search or depth-first search. These two algorithms begin with a random vertex and then searches for the next one until the complete component is parsed. Then another, new vertex is chosen and the next component is analyzed, and so on until every vertex is allocated.

Cliques and Clusters

Clustering measures correlations within the whole network and illustrates how interconnected the nodes are with each other. The local clustering coefficient determines how involved it is in a “clique”. The network average clustering coefficient is the mean of the local clustering coefficients. Local clustering coefficients are described in Section 3.2.3. Another clustering coefficient is the global clustering coefficient. This value is calculated by the number of closed triplets divided by the number of connected triplets of vertices. Triplets are either open triplets (three vertices, connected with two edges) or closed triplets (three vertices, connected with three edges). A triangle consists of three closed triplets. The formulae for these two values are:

$$C_{average}=\frac{1}{n}\sum_{i=1}^{n}C_i \qquad C_{global}=\frac{3*triangles}{connected\ triples\ of\ vertices}=\frac{closed\ triplets}{connected\ triples\ of\ vertices}$$

According to the undirected graph in Figure 10 the global clustering coefficient is:

$$C_{global}=\frac{closed\ triplets}{connected\ triples\ of\ vertices}=\frac{1}{2}=0.5$$

3.2.3 Important Actors in Social Networks

The next step in analysis is to take a look at individual players. Who is important to the network or has a special function? For undirected networks, the measures used are called *centrality* (degree, closeness and betweenness), for directed networks the measures are called *prestige* (i.e. proximity or page rank). With common knowledge, prestige has a positive indicator. Network prestige is not meant positively, it can also take negative meanings and therefore its overall meaning is neutral.

Degree values

Degree centrality illustrates the numbers of links each individual node owns ($C_d(v)$). The degree centrality for directed networks are for both, ingoing and outgoing degrees. To get a standardized degree centrality ($C'_d(v)$) the values are divided by the possible total amount of links.

$$C'_d(v)_{undirected}=\frac{C_d(v)}{|V|-1} \qquad C'_d(v)_{directed}=\frac{C_d(v)}{2(|V|-1)}$$

In the example graph in Figure 10 the degree centrality for node *A* is 1, for node *B* 3 and for node *C* and *D* it is 2. The standardized values are:

Node	$C_d(v)$	$C'_d(v)$
A:	1	0.33
B:	3	1.00
C:	2	0.67
D:	2	0.67

Table 8: Degree centrality of the graph in Figure 10

For directed networks there are two additional degree values, indegree and outdegree prestige. Indegree prestige only considers ingoing arcs, whereas outdegree prestige only respects outgoing arcs.

$$P'_{indegree/outdegree}(v)=\frac{P_{indegree/outdegree}(v)}{|V|-1}$$

Additionally there is a centralization value for the whole network too. For this it is necessary to choose the actor v^* with the highest deegree centrality score. From this centrality score, the degree centrality score of every other node is subtracted and the results calculated. From this value, the number of nodes minus one is divided and multiplied by the number of nodes minus two.

$$C_d=\frac{\sum(C_d(v^*)-C_d(v))}{(|V|-1)(|V|-2)}$$

For Figure 10 the centralization value for the whole network would be

$$C_d=\frac{\sum(C_d(v^*)-C_d(v))}{(|V|-1)(|V|-2)}=\frac{1-0.\dot{6}+1-0.\dot{6}+1-0.\dot{3}}{3*2}=\frac{1.\dot{3}}{6}=0.\dot{2}\ .$$

Closeness centrality

A more complex measure is closeness centrality. This metric expresses how close an individual node is to all other nodes. The intent is to calculate the shortest distance to every other actor in the network. If this value is low, the node is very central, if it is high, the node is very far away from most other nodes. So the inverse of this value is required to get a dedicated measure. To standardize this measure the value is multiplied by the number of all other nodes.

$$C_C(v)=\frac{1}{\sum(path\,distances)} \qquad C'_C(v)=(|V|-1)*C_C(v)=\frac{|V|-1}{\sum(path\,distances)}$$

For the network in Figure 10 the closeness centrality is

Node	$C_c(v)$	$C'_c(v)$
A:	1/5	3/5
B:	1/3	1.00
C:	1/4	3/4
D:	1/4	3/4

Table 9: Closeness centrality of the graph in Figure 10

A closeness centralization value for the whole network is analogous to the degree centralization, but instead of $(|V|-1)(|V|-2)$ the sum is divided by $(|V|-1)(|V|-2)/(2|V|-3)$.

For the graph in Figure 10 this would be

$$C_c=\frac{\sum(C_c(v^*)-C_c(v))}{(|V|-1)(|V|-2)/(2|V|-3)}=\frac{1-0.6+1-0.75+1-0.75}{3*2*1}=\frac{0.9}{6}=0.15\ .$$

Betweenness centrality

Betweenness centrality expresses how many of the shortest paths of other nodes actually go through every individual node. To standardize this measure, it is divided by the possible maximum. The formulae is:

$$C'_b(v)=\frac{2*C_b(v)}{(|V|-1)*(|V|-2)}$$

In the following table, a possible option is depicted, whereby the betweenness centrality ($C_b(v)$) of a node can be manually computed with the example of a simple ring network (Figure 11 and Table 10).

	2	3	4	5	6	7	8
2		0	0	0	½	1	1
3			0	0	0	½	1
4				0	0	0	½
5					0	0	0
6						0	0
7							0
8							

Table 10: Path matrix for node 1 of Figure 11

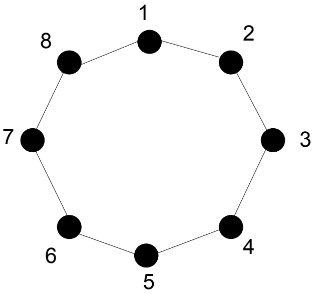


Figure 11: Ring network with 8 nodes

In Table 10 for every lot the shortest path is computed and if this path goes through the node 1, the written value is 1, otherwise it is 0. If there are 2 shortest paths for a node, and only one goes through node 1, the value is 1/2. If there are 3 shortest paths and 2 go through node 1, the value would be 2/3, and so on. Every value will be calculated and the result for node 1 in Figure 11 is $C_b(1)$ = 4.5. In Figure 11 every node is equal, so every node has the betweenness centrality of 4.5.

The betweenness centralization value for the whole network is analogue to the degree centralization, but instead of $(|V|-1)(|V|-2)$ we divide $(|V|-2)(|V|-1)^2$.

Proximity prestige

In Section 3.2.2, prestige was introduced. The Indegree or Outdegree prestige for directed networks is similar to the degree centrality for undirected ones. Hereby, proximity prestige for directed networks is like closeness centrality for undirected ones. The importance of a node depends on the distance of other nodes that can reach it. Many paths with a low path distance give high proximity prestige, less paths with a high path distance gives low proximity prestige. Proximity prestige for an individual node v is computed by counting every node w that can reach v (M_v). The next step is to divide the sum of path distances (S_v) between w and v . To standardize the value the quotient is multiplied by

$M_v/(|V|-1)$.

$$P'_p(v)=\frac{M_v}{S_v}*\frac{M_v}{|V|-1}$$

Rank prestige

Another kind of prestige is rank prestige, which gives weighting to the actors in a network based on their importance. The concept behind this kind of prestige is to see links as votes. Votes from important vertices have a higher weighting than the votes of unimportant ones. Every prestige value an actor possesses will be added to the prestige of every actor connected with an outgoing edge. The first consideration to solve this problem is with arithmetic expression. If we take the example of Figure 12, we can compute rank prestige with these 5 expressions:

Vertex A: $x_A=0$

Vertex B: $x_B=x_C+x_E$

Vertex C: $x_C=x_A+x_B+x_D$

Vertex D: $x_D=x_B+x_C+x_E$

Vertex E: $x_E=x_A+x_D$

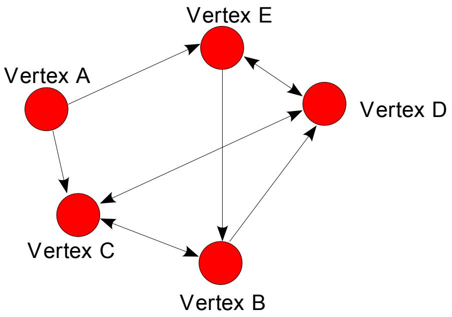


Figure 12: Directed homogenous graph with 2 outdegrees

This system of equations has no other solution than a zero prestige vector, which leads to the formula $A^T x=\lambda x$. A is the adjacency matrix of the graph, x is a vector with prestige values and λ the proportionality factor between rank prestige values and the vote weightings. Here it is hard to find a positive λ that results in a solution that differs from zero vector. x of $Bx=\lambda x$ is known as an eigenvalue of matrix B . So, a positive eigenvalue of A^T needs to be calculated. After this, we solve the system of equations with the calculated value of λ in $A^T x=\lambda x$ and reach a solution. x is an eigenvector.

A modified, simpler solution to this problem is the page rank algorithm. Here a probability matrix P includes the blur value α (normally 0.1) and is used in a continuous algorithm. The first thing that needs to be computed is the probability matrix. For this the adjacency matrix is required, each value of 1 is divided by the numbers of 1's in a row to get P_1 . The next step is to multiply the whole result matrix by 1- α . Finally $\alpha/|N|$ is added to every entry of the result matrix to obtain P . Next, a random starting row x_0 of the matrix P_1 is required, and then $x_0 P=x_1$, $x_1 P=x_2$, $x_2 P=x_3$, and so on needs to be computed until an applicable value is reached. Table 11 shows a sample probability matrix and Table 12 the results of the computation steps.

0.02	0.02	0.45	0.02	0.45
0.02	0.02	0.45	0.45	0.02
0.02	0.45	0.02	0.45	0.02
0.02	0.02	0.45	0.02	0.45
0.02	0.45	0.02	0.45	0.02

Table 11: Probability matrix P

Prestige	A	B	C	D	E
x_0	0	0	0.5	0	0.5
x_1	0.02	0.45	0.02	0.45	0.02
x_2	0.18	0.04	0.41	0.23	0.22
			....		
x	0.00	0.22	0.28	0.33	0.17

Table 12: Page rank computation steps

Local clustering coefficient

The clustering coefficient is, as mentioned in Section 3.2.2, a measure for “cliquiness”. The network average clustering coefficient uses local clustering coefficients to get a metric for the whole network. The local clustering coefficient is a measure for individual nodes.

In order to compute this metric, we take a node v_i and look at the interconnection of its neighbors. If every node related to v_i is connected, the cluster coefficient is 1 (see Figure 13). If none of the nodes related to v_i are connected, the coefficient is 0. The local clustering coefficient from a node v_i, a set of neighbors N_i, and a set of edges E_i between two nodes in N_i, can be computed as follows:

$$LCC_{undirected}(v_i)=\frac{2E_i}{|N_i|*(|N_i|+1)} \qquad LCC_{directed}(v_i)=\frac{E_i}{|N_i|*(|N_i|+1)}$$

The edges E_i are divided by the maximum of possible edges, very similar to the network density in Section 3.2.2. Figure 13 illustrates 4 examples of a local clustering coefficient.

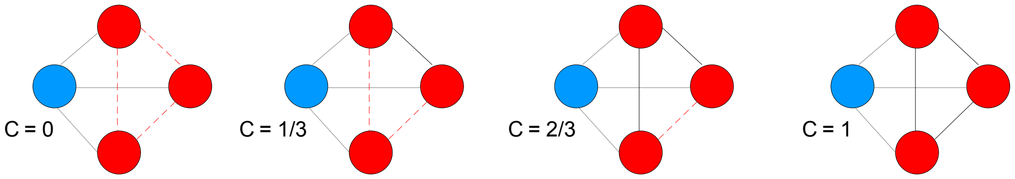


Figure 13: Clustering coefficient example

3.2.4 Groups in the Social Network

Aside from the extraction of network metrics for whole networks and metrics for individual players, it is also possible to detect cliques and groups affiliation. The purpose behind this detection method is to find very dense parts of the networks to gather information about potential groupings.

For an example on groups in the network, Figure 3 on page 10 and Figure 7 on page 13 show the groups and clusters in the network with different colors.

Common cohesive subgroup methods

The easiest way to analyze groups is to find cliques. Cliques are networks or sub-networks with maximum density. Every node is interlinked with every other node. According to this a clique-concept can be relaxed to a k-plex where only the majority of the group or subgroup is interlinked. This method has not been established very well in scientific analysis, with only one notable exception. Figure 14 illustrates a clustering algorithm with k-plex [40].

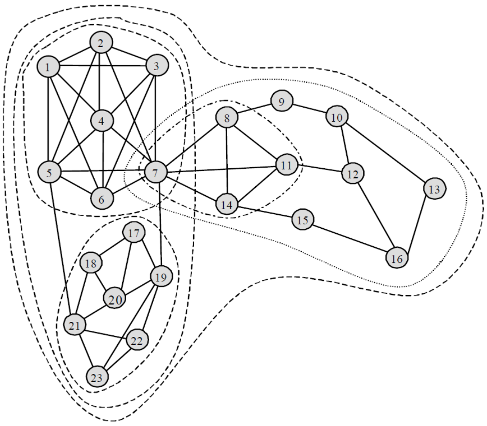


Figure 14: Nested connectivity sets [40]

Community detection algorithms

Community detection algorithms are very frequently used in social network analysis. The most popular is the Girvan-Newman algorithm. This algorithm continuously deletes the node with the highest betweenness. The assumption for this, is that the node with the highest betweenness interlinks two very dense groups [41]. Nevertheless, this algorithm doesn't work well under all conditions and there is a slight arbitrariness to it.

Subsequently, Newman developed more comprehensive methods to find dense parts in the network with even more reliability [42]. Figure 15 illustrates a network of books, where the edges are readers who read the same book. The shapes illustrate the political attitude of the books.

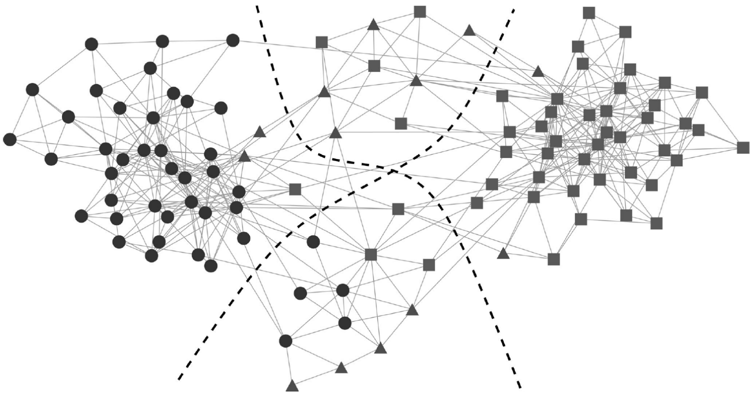


Figure 15: Network communities of books that were read by the same readers [42]

3.2.5 Membership Analysis

In social network analysis we should never forget to analyze the network members. The metrics above treated every vertex equally. Actors, literal authors, communication participants, bloggers etc. are not equal.

One analysis criteria is homophily [43]. Figure 4 on page 11 demonstrates homophily as bloggers of the same political orientation interlink with each other more often than with those of differing political orientation. In this example the homophily would be politics. Another example is bloggers that are more famous who interlink amongst themselves more often than with bloggers that have no fame. A very important question about homophily is not whether it exists, but rather which criteria organizes the network (fame, politics, …).

Another approach is assortative mixing [44]. This theory assumes that nodes tend to connect to other nodes that are like them in some way. If an actor has a high degree value, it tends to connect with other actors of high degree value. If an actor has a low degree value, it connects with other nodes of low degree values.

3.2.6 Additional Network Metrics

There are further metrics which will not be considered as part of this thesis. Without focusing on the details, they are briefly mentioned and described below.

Average path length

The average path length is a metric for analyzing network interconnection. It is the average of all the shortest paths.

Bridges

A bridge is an edge or link whose erasing increases the number of connected components. Depending on the network, the actors of this links can be very important. A local bridge is a bridge whose actors share no mutual neighbors.

Reach

Reach is the degree any member can reach other members of the network with.

Structural equivalence

Structural equivalence states that nodes have common linkage to other nodes in the network. To be structurally equivalent, nodes do not have to be linked with each other.

Structural hole

Linked to the idea of social capital [61], structural holes can be filled with an actor to control the communication of two people or groups.

The research on metrics is far from being completed. Scientists keep the balance between research on metrics and research on social networks and their actors. The line between these two is also very blurred. Sometimes new views and new intentions on social networks lead to new metrics and new algorithms. But it isn't just social network researchers that invent new metrics, there are also many scientists working on biological, chemical or physical network analysis who have already provided and will provide input into social network analysis. For additional literature on Chapter 3 [34] and [35] are recommended.

4 Linked Data

So far, social networks have been introduced, their practical use for observing communication and relationship patterns of human beings, their classification into complete, partial and ego-centric networks, and methods to statistically analyze them. This section presents the second main research field which is relevant to this thesis – Linked Data. Special focus will be given to the technical side of Linked Data in the context of data processing and extraction possibilities.

4.1 About Linked Data

The web is full of countless sites with information. These sites are designed for human consumption. The underlying data is mostly in hidden databases or storage (the so called "*deep web*"). If one wanted to read the information, there would be no difficulty in doing so, however problems would arise when trying to read and parse this data automatically with a machine. Every website is different, so the data has to be parsed out of HTML or other (sometimes proprietary) formats with difficulty.

The movie "Tron" could be used as an example for this. There is informational data about "Tron" on Wikipedia[1], additional information, especially on staff and actors, at the internet movie database[2], and even further additional information, this time especially reviews and the ratings of hundreds of users, can be found on Rotten Tomatoes[3]. There is a lot of redundant data on each server, but even more additional data to complement.

An external machine has problems in finding information on, for example, the date the movie was published, or on the actors of the film. The machine needs very specific knowledge of the individual

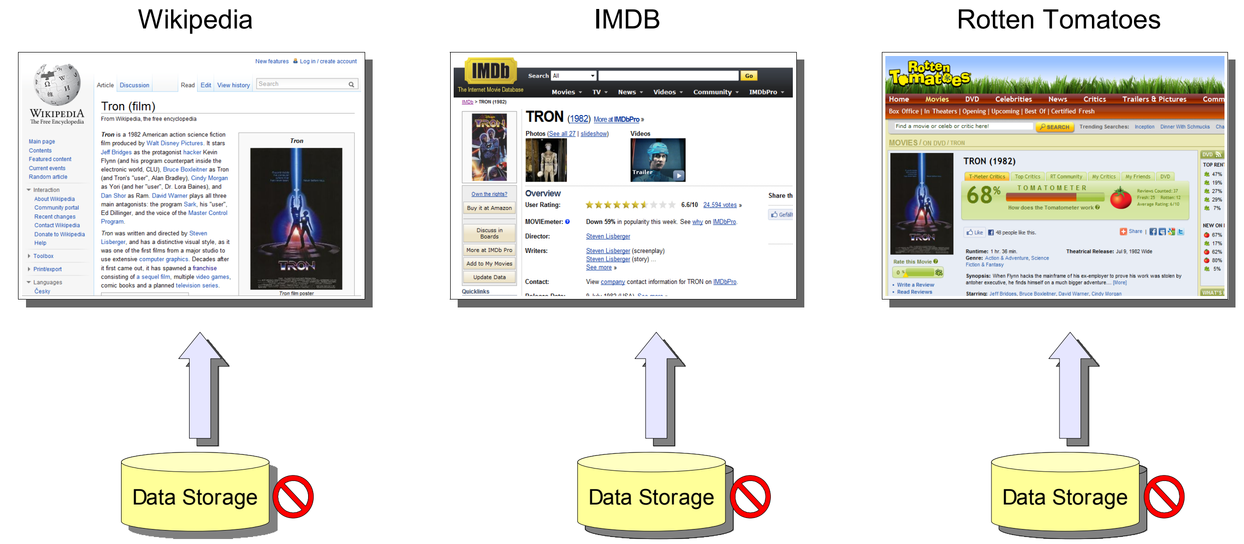


Figure 16: Every website has its own, closed data storage

[1] http://www.wikipedia.org

[2] http://www.imdb.com

[3] http://www.rottentomatoes.com

HTML files, because the information embedded in the file is not well-structured. Figure 16 illustrates the whole situation.

Nevertheless, if we want to automatically extract data from the Web, we would have to use a plugin for at least every website to read the data. This is not applicable. The web would never have had success, if every website would have needed its own plugin. The web is successful because of the uniform standards HTML [47] (for encoding), URI [48] (for addressing) and HTTP [49] (for transportation) for the exchange of hypertext multimedia documents. The vision of the Linked Data [50] movement, a part of the Semantic Web group [51] of the W3C (World Wide Web Consortium) [45], is to apply the same successful concept of uniform standards to the underlying, machine-readable data. The Linked Data principles [50] are:

1. Use *URIs* as names for things.
2. Use *HTTP URIs* so that people can look up those names.
3. When someone looks up an *URI*, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other *URIs*. so that they can discover more things.

Linked Data uses *URIs* for addressing, HTTP for transportation and (unlike the World Wide Web) RDF [52] and related standards for modeling and encoding. Synonyms for Linked Data are (Linked) Open Data, Web of Data or Web of Open Data.

4.2 Enabling Technologies

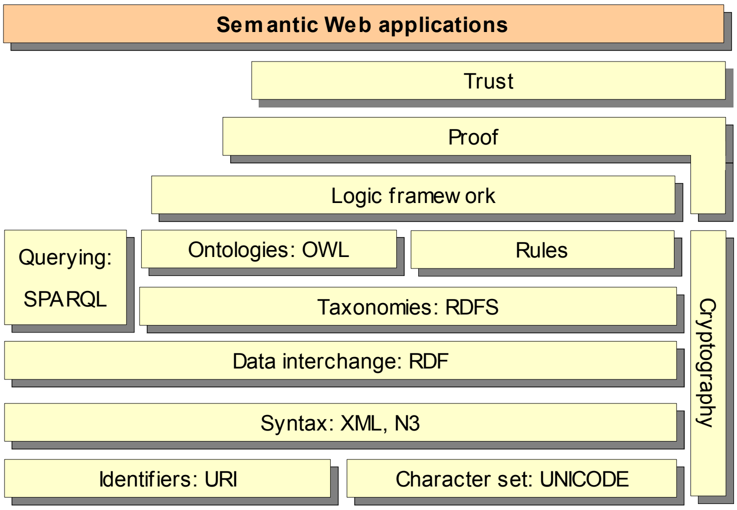


Figure 17: Technologies of the Semantic Web

Figure 17 illustrates the underlying stack of Semantic Web technologies. Actual working groups have finished their research bottom-up to the ontology section and are now improving and working on the logic frameworks and additionally renewing existing concepts (OWL2 [56], RDFa [57], etc.).

In this Figure we can see vocabularies (ontologies, taxonomies and rules) that are based on an RDF data model. This model is serialized in certain formats (XML [55], N3 [46], etc.) and *URIs* can identify most parts of this model.

4.2.1 The RDF Data Model

The data model behind Linked Data is RDF which consists of special directed graph constructs. This data model consists of resources and literals. Resources are *URIs* and literals are text strings. A combination of resource, resource and resource or literal is called an RDF triple. This triple is denoted as a connection between two nodes, *vertex – link – vertex*, or, more often used: *subject – predicate – object*. Subjects and predicates, as mentioned before, are URIs (resources), the object can be an *URI* or a text-label.

Figure 18 illustrates a small exemplary part of the DBpedia resource of the movie "Tron" and its director Steven Lisberger. In this Figure we can see the resource http://dbpedia.org/resource/Tron_(film) with some labels, its title "Tron", its abstract in English and German and its link to the director of the movie. This resource http://dbpedia.org/resource/Steven_Lisberger additionally has all information about the person Steven Lisberger like his name, his SKOS [58] description, etc.. The result is a huge informational graph.

With the RDF data model it is even easily possible to link databases. Figure 18 displays two nodes in different databases that are apparently the same. These are possibly interlinked with a "owl:sameAs" predicate, with a link from DBpedias Tron resource to the node of Freebase's Tron resource.

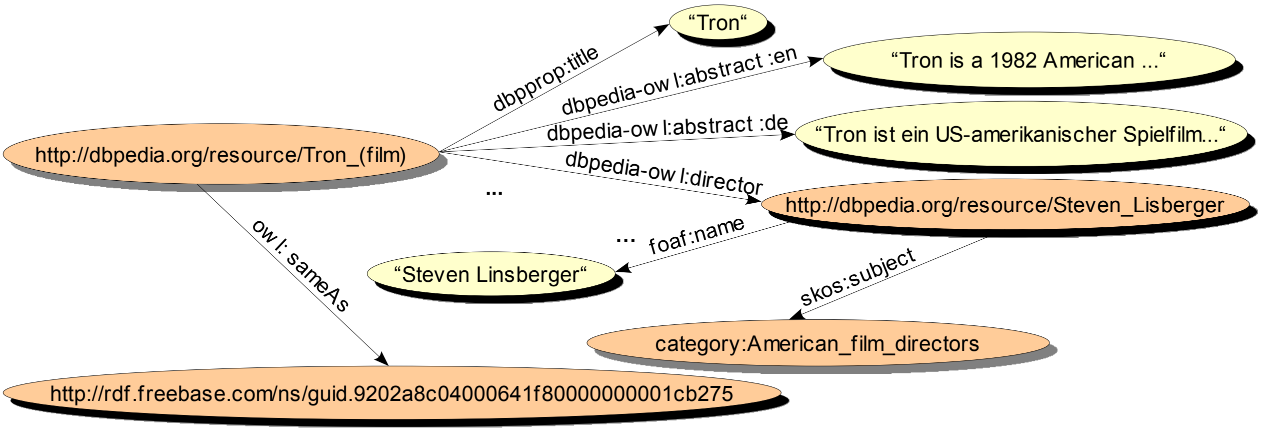

"Tron"
"Tron is a 1982 American ..."
"Tron ist ein US-amerikanischer Spielfilm..."
dbpprop:title
dbpedia-ow l:abstract :en
dbpedia-ow l:abstract :de
dbpedia-ow l:director
http://dbpedia.org/resource/Tron_(film)
http://dbpedia.org/resource/Steven_Lisberger
...
...
foaf:name
"Steven Linsberger"
skos:subject
category:American_film_directors
ow l: sameAs
http://rdf.freebase.com/ns/guid.9202a8c04000641f80000000001cb275

Figure 18: DBpedia data of the movie Tron

4.2.2 Semantic Web Vocabulary

The RDF Description Language, also called RDF Schema (RDFS) [53] is the basic vocabulary for the definition of classes, subclasses, properties, sub-properties, data types, literals and comments, among others.

The Web Ontology Language (OWL) [54] is a language for defining ontologies and vocabularies. It extends RDF and RDFS with complex semantic expressiveness. The vocabulary of OWL includes the above mentioned *owl:sameAs* property. Because of their capabilities and similarity to cognitive relations, RDFS and OWL are also used in artificial intelligence research.

The Simple Knowledge Organization System (SKOS) [58] is a model for representing thesauri, controlled vocabularies, taxonomies, etc.. According to Figure 18, Steven Lisberger is linked via *skos:subject* with *category:American_film_directors*. The SKOS attribute *category:American_film_directors* is linked via *skos:broader* with *category:English-language_film_directors*. So Steve Lisberger is indirect of the attribute *category:English-language_film_directors*.

Friend of a Friend (FOAF) already mentioned in Section 2.5.2 on page 13 is a vocabulary to interlink social software data. FOAF is one of the first Semantic Web ontologies. It is used by social network platforms like LiveJournal, PeopleAggregator, Tribe.Net, etc..

4.2.3 Uniform Resource Identifier

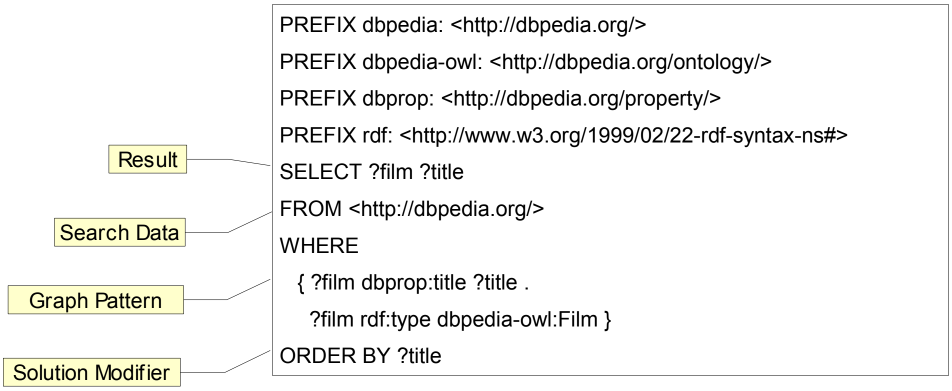
Uniform Resource Identifiers [48] are unique names of RDF resources. Every resource must be identified with an URI. In Figure 18 the orange nodes and the edges are URIs. Some URIs are shortened with a namespace-prefix, like *owl*, *dbpedia-owl*, *dbprop* or *category*. These namespaces can be replaced by their long description. o*wl:sameAs*, for instance is a short description for http://www.w3.org/2002/07/owl#sameAs with http://www.w3.org/2002/07/owl# as namespace. URIs are in most cases accessible via HTTP.

4.2.4 The SPARQL Query language

SPARQL [59] is a query language and protocol to request data from an RDF database. Its structure is similar to SQL. Its request structure is a quadruple consisting of a graph pattern, search data, solution modifier and result.

$$Request=(GP, SD, SM, R)$$

A basic example is described in Listing 3:



```
PREFIX dbpedia: <http://dbpedia.org/>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbprop: <http://dbpedia.org/property/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?film ?title
FROM <http://dbpedia.org/>
WHERE
  { ?film dbprop:title ?title .
    ?film rdf:type dbpedia-owl:Film }
ORDER BY ?title
```

Listing 3: Example SPARQL query

Serialization Formats

A very convenient, slim data structure is Notation 3 (N3). This data structure is very complex and includes the languages N Triples, Turtle, SPARQL Where and even more extras. A basic example of a N3 data structure is listed in Listing 4.

The structure in Listing 4 begins with a declaration of the prefixes, followed by the definition of the triples. Subjects, predicates, objects, are separated with a blank. Does a subject has more predicates and objects, they are declared behind the semicolon separator. If a subject and a predicate has more than

Notation 3 data structure

```
@prefix dbpedia-owl: <http://dbpedia.org/ontology/>.
@prefix dbpprop: <http://dbpedia.org/property/>.
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix category: <http://dbpedia.org/resource/Category:>.
@prefix opencyc: <http://rdf.freebase.com/ns/guid.>.
@prefix foaf:<http://xmlns.com/foaf/0.1/> .
<http://dbpedia.org/resource/Tron_%28film%29> dbpprop:title "Tron"; dbpedia-owl:abstract
"Tron is a 1982 American..."@en, "Tron ist ein US-amerik..."@de;  dbpedia-owl:director
<http://dbpedia.org/resource/Steven_Lisberger>; = opencyc:Mx4rvddV6ZwpEbGdrcN5Y29ycA .
<http://dbpedia.org/resource/Steven_Lisberger> foaf:name "Steven Lisberger"; skos:subject
category:American_film_directors .
```

Listing 4: N3 example of Figure 18

one sequel, for example to declare a text for different languages, the separator is a comma. A dot finishes the declaration of one subject to start another.

Another frequently used data structure is RDF/XML. For XML there is a lot of support and powerful parsers. Listing 5 uses the the model of Figure 18 as example to give comparability to Notation 3 in Listing 4.This structure starts with a root node named “rdf:RDF” with namespace attributes. Subjects are “rdf:Description” nodes with an “rdf:about” attribute with its name. The next nodes are the predicate of the triples with the object as its value.

RDF/XML data structure

```
<?xml version="1.0"?>
<rdf:RDF xmlns:dbpedia-owl="http://dbpedia.org/ontology/"
xmlns:opencyc="http://rdf.freebase.com/ns/guid."
xmlns:category="http://dbpedia.org/resource/Category:"
xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:dbpprop="http://dbpedia.org/property/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:skos="http://www.w3.org/2004/02/skos/core#"
xmlns:damloil="http://www.daml.org/2000/12/daml+oil#">
  <rdf:Description rdf:about="http://dbpedia.org/resource/Tron_%28film%29">
    <dbpprop:title>Tron</dbpprop:title>
       <dbpedia-owl:abstract xml:lang="en">Tron is a 1982 American...</dbpedia-owl:abstract>
            <dbpedia-owl:abstract xml:lang="de">Tron ist ein US-amerik...</dbpedia-
owl:abstract>
            <dbpedia-owl:director>
             <rdf:Description rdf:about="http://dbpedia.org/resource/Steven_Lisberger">
              <foaf:name>Steven Lisberger</foaf:name>
           <skos:subject
rdf:resource="http://dbpedia.org/resource/Category:American_film_directors" />
             </rdf:Description>
            </dbpedia-owl:director>
     <damloil:equivalentTo
rdf:resource="http://rdf.freebase.com/ns/guid.Mx4rvddV6ZwpEbGdrcN5Y29ycA" />
  </rdf:Description>
</rdf:RDF>
```

Listing 5: RDF/XML example of Figure 18

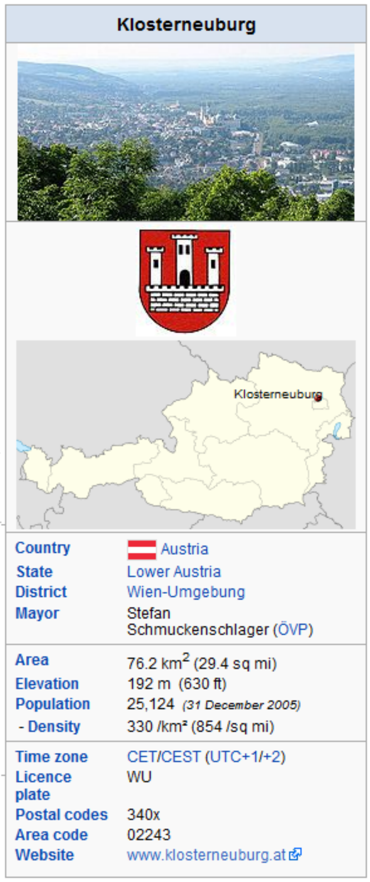
4.3 DBpedia, the Nucleus of the Web of Open Data

Wikis are systems for collaborative authoring, versioning and publishing of texts and images. Wikipedia is the most famous and successful Wiki with the goal to build a large online encyclopedia. The Wikipedia succeeded with its intentions with countless authors on over 10 million articles in more than 250 languages.

The DBpedia[4] project uses the information of Wikipedia templates to semantically interpret and extract information. Figure 19 shows a template of the city Klosterneuburg in Lower Austria. Templates are roughly categorized in [68]:

- Geographic : countries, cities, rivers, mountains,...
- Education: universities, schools, ...
- Plants : trees, flowers,...
- Organizations: companies, sport teams,...
- People: politicians, scientists, presidents, athletes, ...

This data is converted to RDF data and made freely available on the Web. So, DBpedia is the biggest Linked Data project, developed at the University of Leipzig, “Freie Universität Berlin” and OpenLink Software. The first available public data was published in 2007 with free licenses. It is a huge knowledge base for many musicians, politicians, athletes, media, regions, clubs, and many, many more. The dataset consists of 3.4 million “things”, with 312,000 persons, 413,000 places, 94,000 music albums, 49,000 films, 15,000 video games, 140,000 organizations, 146,000 species and 4,600 diseases, among others, and over 1 billion facts (RDF triple). There are abstracts of 3.2 million things in 92 different languages, 841,000 links to images, 5 million links to external web-pages, over 9 million interlinked RDF data to external sources (Yago, GeoData, …).



Klosterneuburg

Country	Austria
State	Lower Austria
District	Wien-Umgebung
Mayor	Stefan Schmuckenschlager (ÖVP)
Area	76.2 km^2 (29.4 sq mi)
Elevation	192 m (630 ft)
Population	25,124 *(31 December 2005)*
- Density	330 /km^2 (854 /sq mi)
Time zone	CET/CEST (UTC+1/+2)
Licence plate	WU
Postal codes	340x
Area code	02243
Website	www.klosterneuburg.at

Figure 19: Wikipedia Template of Klosterneuburg

DBpedia works like a RESTful server with every resource accessible via HTTP. Additionally to the accessible resources, DBpedia provides a SPARQL endpoint. Serialization formats DBpedia can return are RDF/XML, N3/Turtle, JSON+RDF, OData/Atom and HTML.

An existing project on DBpedia is RelFinder [69], a project that searches for paths between two resources in the RDF data model. Nevertheless it is possible to search for a connection between two persons, it is no classical approach for social network analysis. Paths can lead through non-person objects like a common profession or the a place one person was born, the other person died.

Other projects on DBpedia are DBpedia mobile [70], a Linked Data browser for the mobile phone, Search

[4] http://dbpedia.org/

DBpedia [67], and Faceted Wikipedia Search [65] for searching DBpedia data, the DBpedia Query Builder for creating queries for DBpedia, and even more.

According to the headline, DBpedia is a nucleus and a crystallization point for the Web of Open Data. Figure 20 illustrates the importance of DBpedia to other Linked Data data sets and projects.

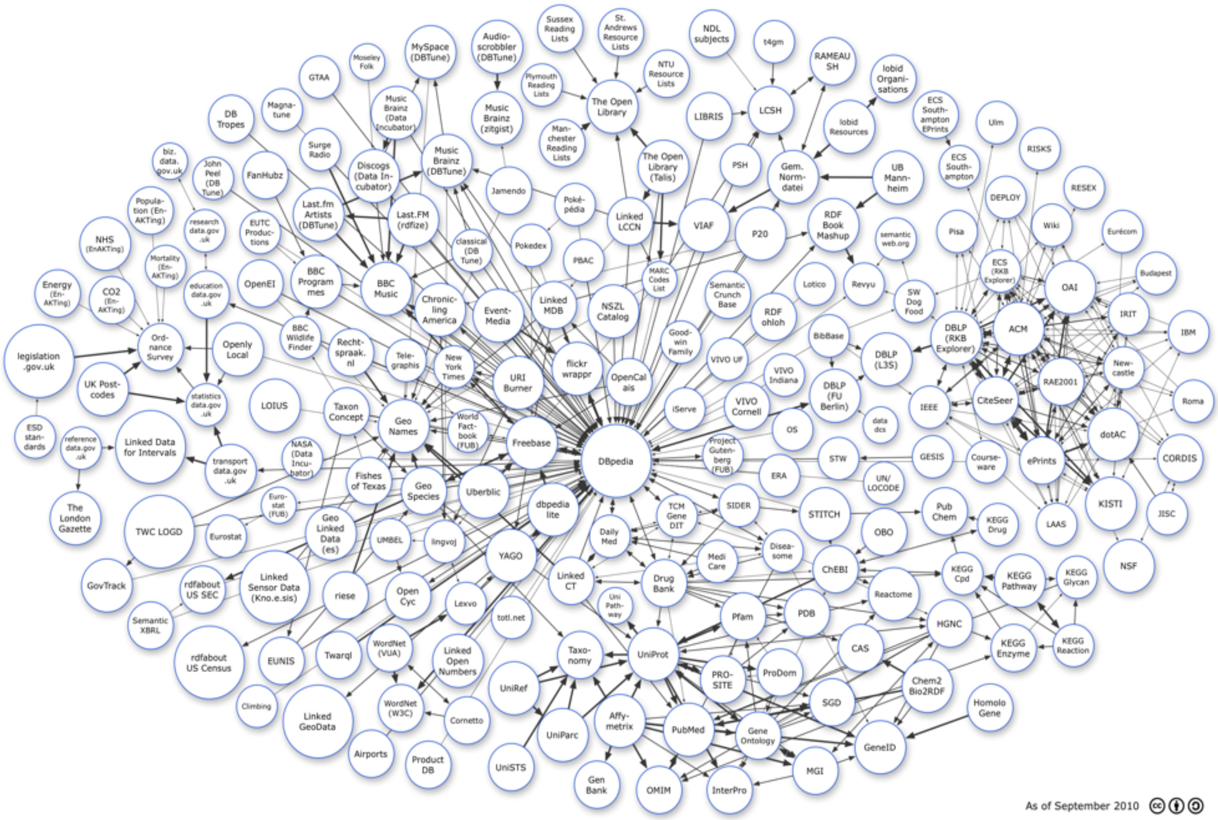


Figure 20: Linked Open Data cloud [60]

4.4 Retrieving Data from Linked Data Sources

There are two possible methods to extract data from Linked Data sources. The commonly used approach is to build a crawler. Such a crawler can go from source to source, using a predefined starting unit, or use an index by using a SPARQL query or a semantic web search engine. The second possibility is to use complex SPARQL queries to extract the needed information. This method has its benefits if we only use a single SPARQL endpoint. The more endpoints, the more complex it will become, because every endpoint has to be found and manually embedded to the extraction system. Another disadvantage of multiple SPARQL endpoints is the possible redundancy of data.

Table 13 displays a matrix of possible extraction methods and data sources on Linked Data.

extraction method data sources	Crawler-based	Query-based
Single data source	Use SPARQL for indexing first, then parse	Use complex SPARQL Queries to get information
Multiple data sources	Use semantic web search engines for indexing first, then parse	Know all SPARQL endpoints to send complex queries to get information

Table 13: RDF extraction matrix

The main approach for social network analysis in the Web of Data is to use existing FOAF profiles. To create social networks out of FOAF profiles [28], best practice is to use *foaf:Person* for nodes and the *foaf:knows* predicate for the edges. The extraction of FOAF consists of 3 steps. The fist step is to discover the instances of *foaf:Person*, then to merge information about unique individuals and finally to interlink the persons with the *foaf:knows* attribute. To merge unique individuals, the most evident option is to compare *foaf:mbox*, *foaf:homepage*, *foaf:name*, *foaf:nick* or *foaf:phone* to analyze unique individuals out of *foaf:Person* data.

This thesis is about analyzing the possibility of extracting social network data on other Semantic Web's Linked Data sources. For this, we decided to take a single, extensive source and work with complex SPARQL queries.

Our source is the data of DBpedia, to be more exact the 312,000 persons illustrated in DBpedia. The goal for the next part of this thesis is, among others, the creation of a framework that extracts social networks out of a single Linked Data source via SPARQL endpoint with complex SPARQL queries. Additionally we illustrate the methodology and conception of the approach and finally the results we established.

Part 2

Methodology, Implementation and Proof of Concept

5 Methodology and Concepts

In the first part of the thesis, social network research, the distinction between communication and relationship networks and different network analysis methods were introduced. Afterwards, theory of the network analysis with an introduction to graph theory and graph analysis, was depicted and finally an insight on Linked Data, their concepts and structure, the most important data sets and vocabularies for social networks and a theoretical section of network extraction, was given.

The second part specifies the main content of this thesis – the extraction of social networks from the Linked Data source DBpedia, the analysis of the results and the investigation of the useability for social science research. In this part, concepts and methodology with a detailed technical implementation and respectable results of this approach will be presented.

This chapter figures the goal of the thesis with its main concepts and deliberations.

5.1 Goal

The goal is divided into three parts. The first one is the creation of a framework for extracting RDF data from single sources. The source can either be a single online SPARQL endpoint or a RDF dump file. The framework should be generic and applicable for any SPARQL-accessible data set, not only for DBpedia.

The second part is the extraction of four meaningful social networks from DBpedia for an accurate analysis. Depending on a preliminary selection, the networks are of historical writers with a directed “influencing” relationship, a network of scientists, to get to know who was the mentor of whom and a network of football players who played at the same three teams as threshold for a relationship. Within this network, we want to analyze if there is a specific dominant nationality, among others. The last network concentrates on architects and the buildings they created.

The third part of the goal is to get feedback from social science students on the usefulness of social network extraction on DBpedia for scientific research.

5.1.1 Application

The application should give the possibility to extract and analyze social networks from linked data sources. Dedicated networks are extracted from RDF sources of single Linked Data data sets or uploaded RDF dumps. The data is analyzed for important metrics and converted into a specific format for deeper analysis with expert tools.

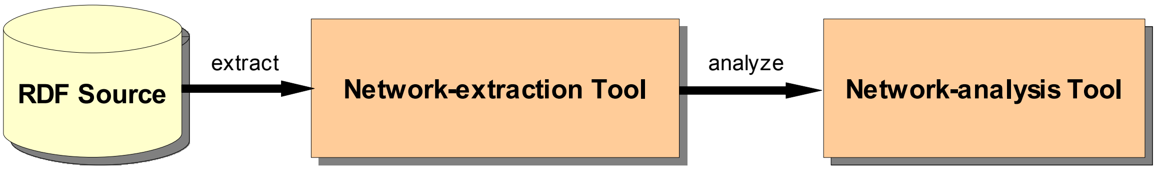


Figure 21: processing steps of an RDF source

The intention of this application is the creation of a web application that connects to single SPARQL endpoints or loads single RDF/XML files in order to analyze their usefulness on social network research. Nevertheless, this thesis is only about extracting data from DBpedia.

5.1.2 Usability

The investigation on usability is carried through with the social network analysis of chosen DBpedia sources. This part follows the question: Are extracted social networks from DBpedia comprehensively analyzable?

A preliminary selection led to four networks that should be extracted and analyzed. A network of writers, a network of scientists, a network of soccer-players and a network of architects. The analysis metrics depend on the specification in Section 3.2 on page 22.

First of all, the network should be visualized to get an overview on it. Secondly, depending on the questions the network is created for, the focus lies on results of the main metrics of the entire network, individual nodes or specific connected components.

5.1.3 Usefulness

At last, advanced knowledge on the usability should be found. We want to know whether our approach is also applicable to the field of social science research. Therefore a survey with social science students should be made, with questions concerning specific topics. One topic focuses on the standing of social network analysis in social science, another one concentrates on Linked Data in general so that we can put their further answers in relation to their knowledge. Actually, it would be fine to get feedback on the extraction and analysis tools, the possibilities on research in this area and the overall feeling concerning this whole approach.

5.2 Application Overview

Social networks should be extracted from Linked Data sources and then analyzed. As mentioned before, we use an extraction method for single SPARQL endpoints to send queries to get nodes and edges. As shown in Figure 22, the result is processed to a graph and analyzed in the main metrics to give an overview on the network and it's actors.

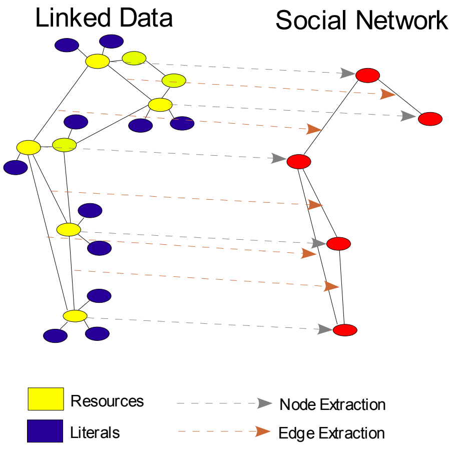


Figure 22: Extraction from Linked Data

5.2.1 Data Flow

Figure 23 depicts a simplified data-flow model. The procedure starts with an input at the web application where the user can ask for a new network. This request is converted to a set of SPARQL queries that are processed to a predefined SPARQL endpoint. The results are converted to a network, a list of nodes and a list of edges. The network is analyzed and stored. The analysis results are transferred to the web application and processed, so the user can view the results.

If the user asks for an existing network, it is easily loaded from the storage. To view single components of existing networks, the system either has to process the component out of the saved complete network, or it is already saved and just needs to be loaded.

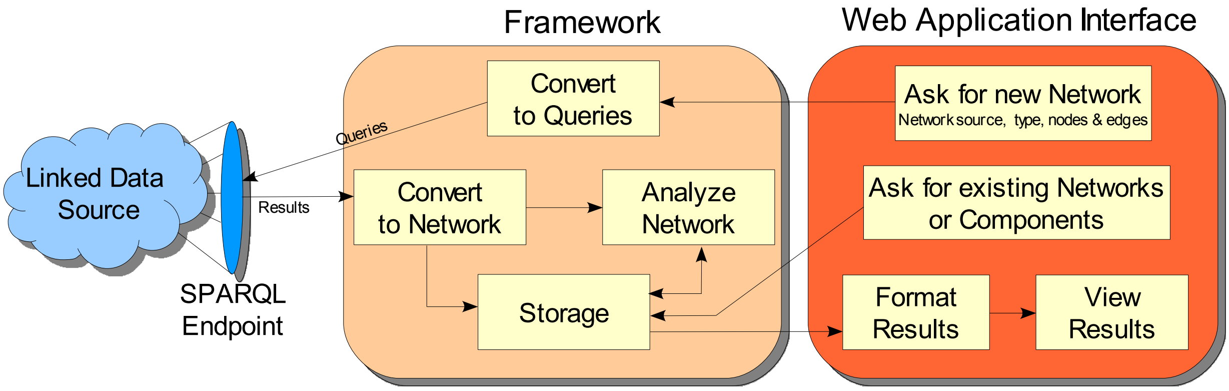


Figure 23: Simplified data-flow diagram

5.2.2 Framework

The framework, with the name SocioCatcher, takes care of all network related processing steps. If the source is a RDF/XML file, the framework has to take care of it and create a small RDF storage on its own. If the source is a SPARQL endpoint, the framework connects directly to it and extracts and computes with complex queries a list of nodes and a list of edges. The framework has to take care of different network types and different edge directions. The following analysis consists of all relevant metrics, depending on the network type and network direction, for a detailed first impression on the network. The framework offers metrics for the whole network, for its nodes and gives a possibility to analyze every connected component separately with these metrics. For further analysis, the generated network is convertible into the universal Pajek-Net format.

5.2.3 Web Application

The purpose behind the web application is to regulate the framework with easy controls, so that the social science students can use the tool. The website should have the following features:

- information about the subject
- extracting networks with an input-form
- present network analysis results
- download networks
- permanently safe networks for sharing them with others
- a questionnaire for the usability study

An important aspect is on the optical web-page design to give a good first impression. Another important design aspect is represented by the input forms and the analysis website, to get competent results from the usability study.

5.3 Network Analysis procedure

The network analysis procedure takes three steps within this thesis: network visualization, the analysis of the metrics and a background investigation on the nodes.

Network visualization

Network visualization helps to get a first impression. The application allows to download the created networks to use professional tools for visualization. Figure 24 illustrates an example for visualization. This network is about musicians, that have a relationship, when they were on the same two record labels.

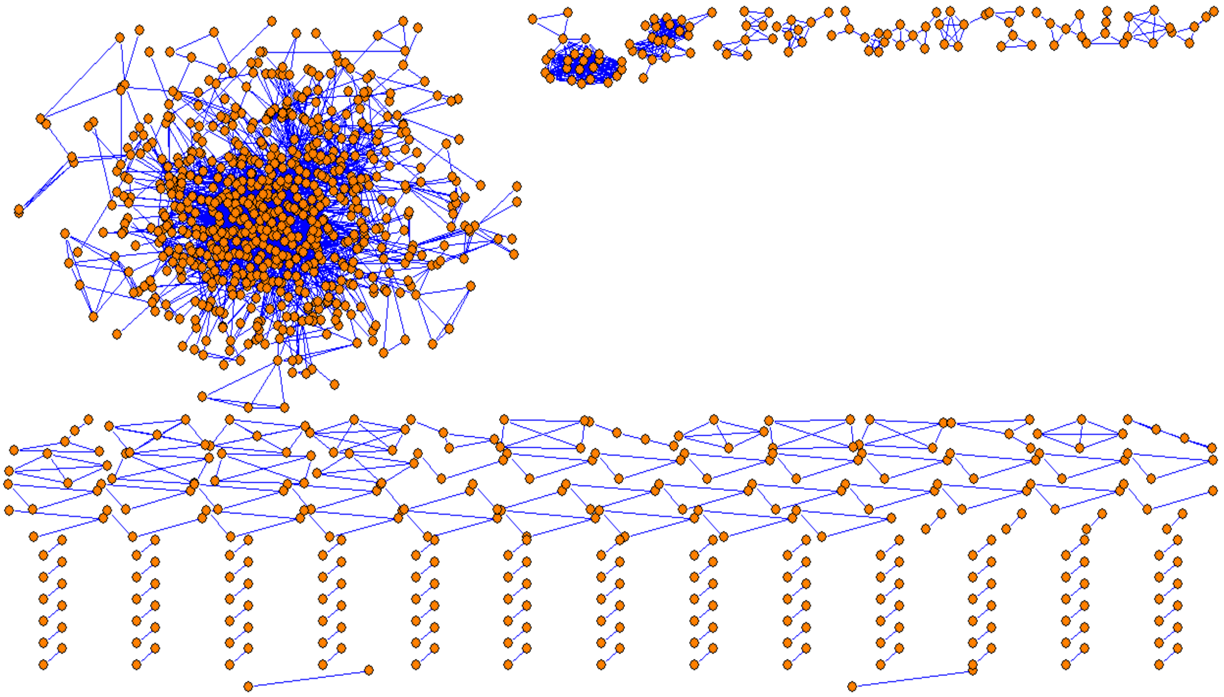


Figure 24: Visualized network of musicians at the same two record labels with Pajek

Statistical Analysis

The main part of the analysis process is represented by the statistical analysis. Metrics are the amount of nodes and edges, as well as the network density and the number of connected components. For undirected networks, metrics on the nodes are degree-, closeness- and betweeness centrality. For directed networks, metrics are indegree-, outdegree- and rank prestige. Additional metrics like proximity prestige can be computed via Pajek.

Background investigation

A background investigation on the network members helps to affirm and find specific patterns of the network or its components.

These three steps are processed for the entire network and also, if needed, for the connected components. There are also no strict separations of these steps, because we want to undertake our questions carefully. The goal is to receive assumptions, theories and patterns on social networks to confirm the usability. For more details on social network analysis, remember Section 3.2 on page 22.

If we can find patterns and statements on the social networks, the extraction and analysis tool, ergo, the whole approach of the technical work, will be investigated as usable.

5.4 Usability Study approach

The usability study contributes to the allocation of usefulness of the application. The questionnaire puts its focus on open questions, but also uses closed and multiple choice questions.

5.4.1 Target Group

The target group for this study are 7 students of sociology at the University of Vienna.

To get a recall of 7 we sent an invitation within the request for taking part in the study to 25 students. The students were randomly chosen by former course contacts of a sociology student, we had an initial expert interview on social science and social network analysis [Appendix A].

5.4.2 Process flow of the Study

The first thing participants had to do, was reading the supplement of the usability study [Appendix B].

The supplement consists of

- an overview on Linked Data and social network analysis,
- information on the web site's extraction and analysis tool and
- three concrete usage scenarios for the extraction tool.

The overview on Linked Data and social network analysis should introduce the thesis related topics to the participants. After reading it, even if he or she has never heard of network analysis and/or Linked Data before, they should know the basics and what this application is all about.

The information on the web site's extraction and analysis tool should be introducing all features. First of all, this part should give information on what's possible with the extraction tool and how to use it, secondly the part gives an overview of the analysis metrics.

Finally, two concrete usage scenarios give step by step instructions to network creation. The first network is a complete network of influenced writers. The second network is an affiliation network of musicians at the same record label.

After completing the usage scenarios, the participants were requested to fill out the questionnaire.

5.4.3 Questionnaire

The questionnaire features a combination of closed questions, specific questions and open questions. Because this is a qualitative study, we hoped to receive the best answers with the open questions. The other question-types complement the questionnaire to ease intensity and detect the overall personal relation to the topics.

The questionnaire was divided into three parts:

Part 1: Knowledge of and attitude towards social network analysis

This part consists of 6 questions with the intention to get to know the opinion on social network analysis of the participant to put the questions of part 3 into the right context. All 6 questions influence the quality of questions on the social network analysis method.

Question 1: Did you know the methodology of social network analysis before this study?

Question 2: Is social network analysis with current paradigms applicable?

Question 3: Network analysis is ... (multiple choice and essay)

Question 4: Is social network analysis with actual questions convenient in social science?

Question 5: What do you think of computational social science?

Question 6: Should there be more interdisciplinary collaboration at the University of Vienna?

After analyzing these questions, we should be able to know about the participants knowledge and opinion on social network analysis and computational social science.

Part 2: Knowledge of and attitude towards DBpedia and Linked Data

This part consists of 7 questions. The intention of this part is to receive information on the attitude of DBpedia, Linked Data, Internet and technical knowledge in general of the participant. Here we find a set of questions for loosening the strictness in the questionnaire and to get a focus on the following part. These questions are of less use for the thesis.

Question 1: On a scale from 1 to 10, how would you rate your technical knowledge?

Question 2: Did you know DBpedia before this study?

Question 3: Do you think DBpedia is a good data source for extracting (historical) social networks?

Question 4: Do you think there are enough fields of research on the web in social science?

Question 5: Do you use social software?

Question 6: Do you think social social software is an interesting field of research in social science?

Question 7: How do you think about privacy in social software?

With question 1 we should know about the technical self-assessment, which influences the quality of questions on DBpedia. Question 3 is a core-question which will be asked in part 3 anew with small changes to check consistency. Questions 4 to 7 are loosening questions on social network related topics on the web to strengthen the answers of part 3. Additionally these questions give further guidance on the technical know how of the participant.

Part 3: SocioCatcher Usability

This part is the most important in this questionnaire. It consists of 5 questions:

Question 1: What do you think of automatically extracted data of DBpedia or other Linked Data sources?

Question 2: Do you think, this website for extracting networks of DBpedia is usable for research in social science?

Question 3: What do you think of the extraction tool on this website?

Question 4: What do you think of the analysis interface on this website?

Question 5: Which aspects do you find ought to be improved?

Question 1 is a closed question to get the perception of the participant of our approach. Question 2 is also closed, with 5 possibilities to check whether the method and/or the raw data is useful or useless for research in social science, our main question on this part of the thesis. Question 3 and 4 is closed, with an open answer field, if the participant feels the part of the website is lacking in some aspects. The last question offers the participant the opportunity to give a concluding feedback in general.

The answers of the questionnaire gives information on how useful our tool will be, in combination with the selected data source, on social research.

6 Implementation

This chapter contributes to the framework SocioCatcher, its features and how it processes and converts the data, and the Web application, it's development steps and technical implementation.

6.1 SocioCatcher Framework

The name of the class SocioCatcher is a combination of the words sociomatrix and dreamcatcher. The class takes care of all network processing steps. As seen in Figure 21 on page 45 it settles between the website, with an user interface for the end-user, and the plain RDF source. It has an interfaces to a MySQL Database, the file system, a JAR (Java Archive) file for the advanced analysis procedure and includes ARC PHP[1] classes. The framework itself is programmed in the language PHP, which is a recursive acronym for "**P**HP: **H**ypertext **P**reprocessor".

6.1.1 Framework Features

The framework extracts, analyzes and even stores networks. There are three possibilities to create an instance of the class SocioCatcher:

- Create a new network
- Load a temporaryly saved network
- Load a permanently saved network

Input

The input for creating new networks is a complex issue. The extraction source, network type, network direction type, the nodes and edges definition and an option whether or not we want to include lonely nodes (nodes without an edge) have to be defined.

The value of the extraction source could be of an URL to an existing SPARQL endpoint or a destination point of a locally saved file. The network type defines if the network is complete, ego-centric or an affiliation network (see Section 2.7.4 on page 16). The network direction type specifies whether the network is directed or undirected. The declaration of the nodes and edges are listed in Table 14.

Table 14 describes the array structure of the nodes and edges definition. With undirected complete networks, nodes are described with one predicate and object. For example, we want "ALL rdf:type dbpedia-owl:Writer". If there is more than one description type, for instance writers and artists, the array is appended by another predicate and object. The edges are described by the predicate.

Additionally to the predicate, directed complete network edges have the network direction, defined by a string with value "left" or "right" appended.

[1] http://arc.semsol.org/

The nodes of ego-centric networks are defined by the ego node, while the the edges are computed by its predicate(s) and, in case of a directed network, the belonging direction.

The nodes of an affiliation network are defined by their common predicate and object, just like undirected networks. The edges are computed by a predicate which directs to an event or container, as mentioned in Section 2.7.4 on page 16. For example we take "ALL rdf:type dbpedia-owl:SoccerPlayer" as our nodes and their relations are depending on the same containers they have with "<<SoccerPlayer defined before>> dbpprop:club CONTAINER". At the end of the edge-definition array a threshold is attached. Afterwards the data is computed and transformed to a social network as described in Listing 1 on page 16.

	Undirected complete network	**Directed complete network**	**Undirected ego-centric network**	**Directed ego-centric network**	**Undirected affiliation network**
Node:	<predicate> <object> [<predicate> <object>]	<predicate> <object> [<predicate> <object>]	<ego>	<ego>	<predicate> <object> [<predicate> <object>]
Edge:	<predicate> [<predicate>]	<predicate> <dir> [<predicate> <dir>]	<predicate> [<predicate>]	<predicate> <dir> [<predicate> <dir>]	<predicate> [<predicate>] <threshold>

Table 14: Nodes and edges definition for the framework input

After network creation, the network receives an identification number. In order to load temporary saved networks, an instance of the class SocioCatcher only needs to know the network identification number and the network type. For permanently saved networks, the class is instantiated automatically only with its identification number.

Output

The output metrics can refer to the whole network that is created or loaded, or to a specific component only. Analysis metrics are:

- Amount of vertices
- Amount of edges
- Network density
- Amount of connected components (only for the entire network)

In case the network is undirected

- Degree centrality for each node
- Closeness centrality for each node
- Betweenness centrality for each node

In case the network is directed

- Outdegree prestige for each node
- Indegree prestige for each node
- Rank prestige for each node

The results of the three additional metrics, that are depending on the direction type of the network, are represented in an array with its index as the index of the nodes. Additionally the framework creates a simple Pajek file for further analysis either for the entire network or for single connected components.

6.1.2 Framework Processing Steps

Figure 25 gives a detailed overview of the interaction of the individual framework parts and the overall data flow in the framework. In this section we take a closer look at the steps, illustrated in Figure 25.

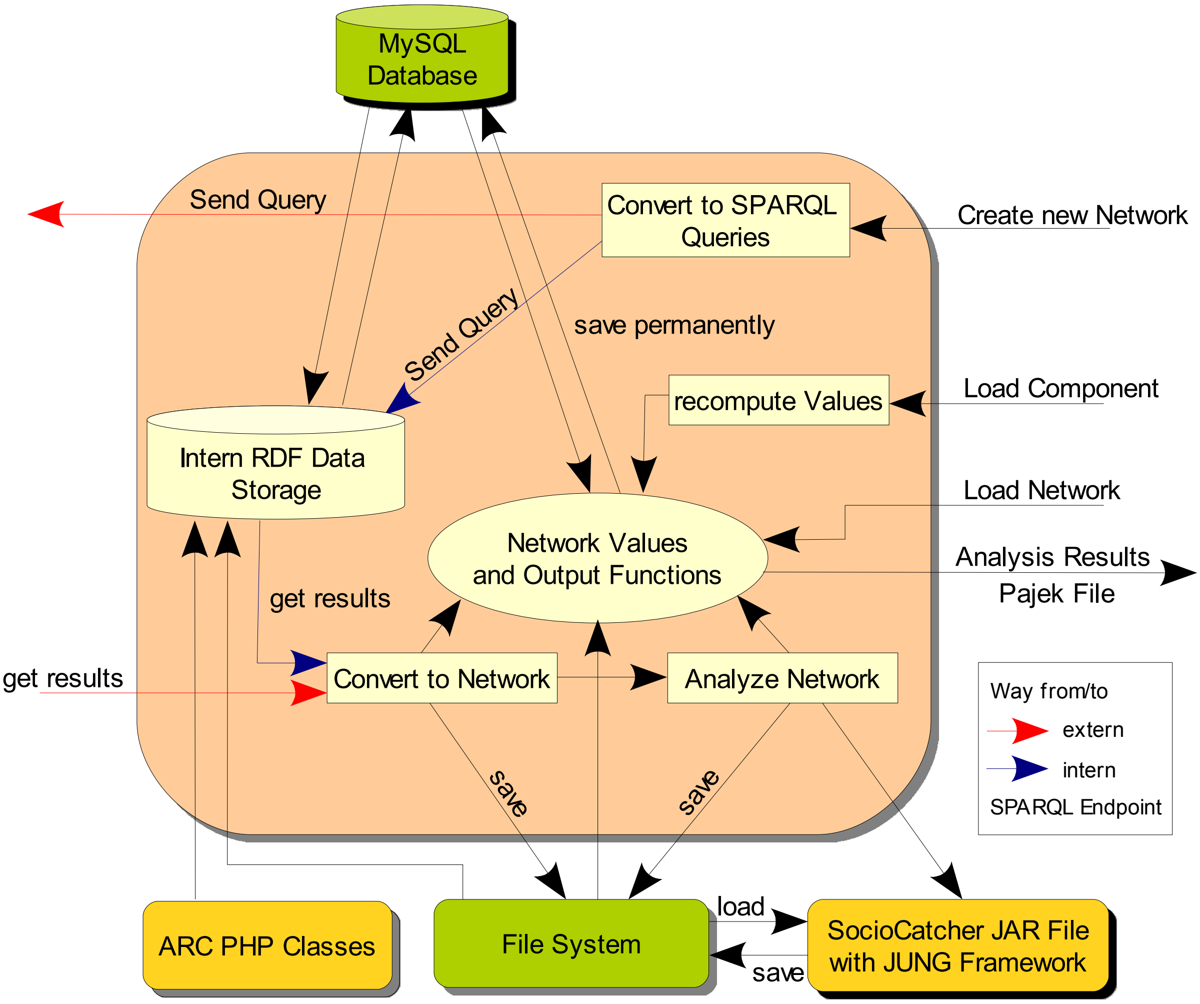


Figure 25: Detailed data flow of the framework

The framework shows three different inputs. The first input is “Create new Network”. As mentioned in Section 6.1.1 this input has to commit the extraction source, network type, network direction type, the nodes, the edges and a flag, if the network extraction should include lonely nodes or not. Depending on this flag, nodes and edges are extracted separately, or only edges are extracted and nodes computed from the result.

SPARQL Query Generation

Depending on the extraction type, there are different possibilities for query building and node extraction. As described in Section 6.1.1, ego-centric networks need a different input for nodes than complete or affiliation networks. With complete and affiliation networks, nodes are extracted by a triple: *?n <input> <input>.* If there is than one node-definition, these triples are connected by an UNION. Listing 6 figures a SPARQL query that returns all *rdf:type dbpedia-owl:Writer* and all *rdf:type dbpedia-owl:Artist*.

```
SELECT DISTINCT ?n
WHERE
 { {?n rdf:type dbpedia-owl:Writer}
     UNION  {?n rdf:type dbpedia-owl:Artist }}
```

Listing 6: SPARQL query for node extraction of writer and artists

For building queries and extracting nodes of ego-centric networks we need to know the definition of the edges. If we want to extract a personal network of Aristotle, we define the edges like: *?n dbpprop:influences ?x* with *?n* is http://dbpedia.org/resource/Aristotle. In addition, to get all other belonging nodes, we take the term {?n <edge> ?x} UNION {?x <edge> ?n} with ?x as our source for Aristotle. Listing 7 illustrates the SPARQL query of this example.

```
SELECT DISTINCT ?n
WHERE
{ { ?n dbpprop:influences ?x filter regex(?n,'http://dbpedia.org/resource/Aristotle') } UNION
{ {?n dbpprop:influences ?x} UNION {?x dbpprop:influences ?n}
         filter regex(?x,'http://dbpedia.org/resource/Aristotle') } }
```

Listing 7: SPARQL query for node extraction of an ego-centric network of Aristotle

The query building for the edge extraction is slightly more difficult. Ego-centric edges are extracted very similar to the nodes. The edges from the ego-node are needed (*?e1 dbpprop:influences ?e2* where *?e1* is ego) and to the ego-node (*?e1 dbpprop:influences ?e2* where *?e2* is ego). Additionally the edges between the alter-nodes (nodes in ego-centric networks that are not ego) are needed, *?e1 dbpprop:influences ?e2* where *?e1 dbpprop:influences <ego>* and *?e2 dbpprop:influences <ego>*. A complete example of an ego-centric network of Aristotle is illustrated in Listing 8.

```
SELECT DISTINCT ?e1, ?e2
WHERE
{ { {?e1 dbpprop:influences ?e2} filter regex (?e1, 'http://dbpedia.org/resource/Aristotle') } UNION
  { {?e1 dbpprop:influences ?e2} filter regex (?e2, 'http://dbpedia.org/resource/Aristotle') } UNION
  { {?e1 dbpedia-owl:influencedBy ?e2} . {
      {?e1 dbpprop:influences ?x} UNION {?x dbpprop:influences ?e1} .
      {?e2 dbpprop:influences ?x} UNION {?x dbpprop:influences ?e2}
    } filter regex (?x, 'http://dbpedia.org/resource/Aristotle') } }
```

Listing 8: SPARQL query for edge extraction of an ego-centric network of Aristotle

Building queries for the edges of complete networks depend on the committed predicate(s) and the node-definition (the predicate and object from the node definition). *?e1 <predicate> ?e2* with *?e1* and *?e2 <node-definition-predicate> <node-definition-object>*. Listing 9 shows an example of a writer and artist network with 3 different influencing attributes.

```
SELECT DISTINCT ?e1, ?e2
WHERE
{ { ?e1 rdf:type dbpedia-owl:Writer } UNION { ?e1 rdf:type dbpedia-owl:Artist } .
  { ?e2 rdf:type dbpedia-owl:Writer } UNION { ?e2 rdf:type dbpedia-owl:Artist } .
  { ?e1 dbpprop:influenced ?e2 } UNION { ?e1 dbpedia-owl:influencedBy ?e2 } UNION { ?e1 dbpprop:influenced ?e2 } }
```

Listing 9: SPARQL query for edge extraction of a complete network of writers and artists

To build an extraction query for the edges of an affiliation networks, the actors *?n* and the predicate which connects the actors with the events/containers *?c* are needed. First we have to define the actors, just like described in the node extraction and add *?n <predicate> ?c* to the query to get a list of all actors and their allocated container. Listing 10 illustrates a SPARQL query for edge-extraction of musicians and their labels.

```
SELECT DISTINCT ?n, ?c
WHERE
{ { ?n rdf:type dbpedia-owl:MusicalArtist }  .
  { ?n dbpedia-owl:recordLabel ?c } }
```

Listing 10: SPARQL query for edge extraction of an affiliation network of musicians

After extracting a list of relations between actors and containers we have to process the data with the algorithm illustrated in Listing 1 on page 16. Afterwards, the edges are filtered with a defined threshold.

RDF Source

The framework has to address an external SPARQL endpoint or has to load and compute a locally saved file with the ARC2 framework. While the results of the locally saved file are read automatically by the ARC2 framework, the external SPARQL endpoint has to return the results in XML format for the framework to compute and transform them accurately.

Network Datastructure

The results of the SPARQL endpoints are converted to a list of nodes and a list of edges. We chose this kind of data structure in order to ease the conversion to the Pajek-Net format and to process the data efficiently. The nodes are saved in a simple array[1] starting with index 1. The edges are stored in a two dimensional array with *array[edge-index] [0]* and *array[edge-index] [1]* containing the indices of the values of the nodes.

After conversion, the results are included in the class structure, saved to a file and sent to the analysis.

Network Analysis

After the creation of a network, the framework computes the amount of nodes, the amount of edges, the network density and the connected components.

Connected components are computed by using a “to-do-list” which contains all nodes, a random start node, and a simple breadth-first search algorithm which searches through the nodes and deletes them from the to-do-list. When the search is finished, another node from the to-do-list is taken. This is repeated until the to-do-list is empty. Finally the result is saved to a file with the index of the component and the index of the node with a blank in between per line.

[1] PHP treats arrays very similar to lists.

If the network is a complete or affiliation network, there are three additional metrics for each node, depending on the direction type of the edges. If the edges are directed, in-degree, out-degree and rank prestige are computed for each node. If the edges are undirected, every node is computed for degree, closeness and betweenness centrality. These metrics are computed by an external JAR file which uses the JUNG[2] (Java Universal Network/Graph) framework. The interface between the PHP framework and the JAR file is a simple *exec* command. With this command, the JAR file loads the network from the file system, computes the metrics for each node and saves these metrics in return to the file system, where the SocioCatcher framework can load them.

Load Network

A network can be loaded by its identification number. If the network is permanently saved, there are additional values: a network name, a network description and the name of the user. A permanently saved network receives an entry in the MySQL database, which is usually used for the ARC2 framework. Figure 26 illustrates the Entity Relationship-diagram in IDEF1X notation.

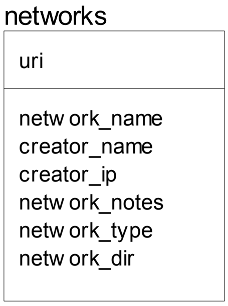


Figure 26: Entity Relationship diagram of permanently saved networks

6.1.3 Framework Variables and Functions

	Identification Variables	Technical Variables	Statistical Variables
public	$uri $networkName $networkNotes $creatorName	$networkType $networkDir $nodes $edges	$nodesCount $edgesCount $networkDensity $conComp
private		$sparql	

Table 15: Class-variables of SocioCatcher framework

$uri is the network identification number. On standard configuration, it is a random string of 15 characters. *$networkName*, *$networkNotes* and *$creatorName* are strings which have to be defined for permanently saved networks. *$networkType* defines the extraction type of the network. Its values can be "complete" for complete networks, "ego" for ego-centric networks or "affiliation" for affiliation networks.

$networkDir defines the network direction type. The value of *$networkDir* can either be "directed" or

[2] http://jung.sourceforge.net/

“undirected”. *$nodesCount* is the amount of nodes in the network. *$edgesCount* is the amount of edges of the network. *$networkDensity* is the network density, computed as mentioned in Section 3.2.2 on page 23. *$conComp* is the amount of connected components the network possesses. *$nodes* is an array of nodes with the node index as array index (so the array starts normally with 1 instead of 0) and the node name as value. $edges is a two-dimensional array with an edge index in the first dimension and the two indices of the nodes in the second dimension of the array. *$sparql* is the location of the SPARQL endpoint. If the source is a file, this value is set to “intern”.

	Network Controls	**Statistical Functions**	**Support Funktions**

public	__construct($uri_or_type, $nwt="savedDB", $nwd="", $location="")
	extractNetwork($nodes, $edges, $no_lonely_nodes=true)
	saveNetwork($nwName, $nwNotes, $cName, $cIP)
	getComponent($identifier)
	getPossibleEdges($nodes)
		getMetric($type)
		getAltersCount()
			loadFromDB()

private	loadFromPajek($filename="self")
	dump($file)
		basicAnalysis()
		advancedAnalysis()
		connectedComponents($myNodes= "self", $myEdges = "self")
			createQueries($node,$edge)
			XMLSPARQLResults($query)
			dbSetUp()
			randomName($nameLength)

Table 16: Class-functions of SocioCatcher Framework

We restrict our detailed description on the functions displayed in Table 16 only to the ones that are public.

Because PHP does not allow function overloading, the constructor is a complex construct and hast one to four parameters. If a new network is instantiated, *$uri_or_type* is set to “sparql” or “dump”, depending on the extraction source. The values of *$nwt* and *$nwd* are sent to the class-variables *$networkType* and *$networkDir* directly. *$location* depends on *$uri_or_type* and is either the URL of the SPARQL endpoint or the file position. When the framework should load a temporary saved network, the class has to be instantiated with a network identification number and the right network type. If a permanently saved network is loaded, the constructor needs only the network identification number.

For extracting a network, the function *extractNetwork($nodes, $edges, $no_lonely_nodes)* is called. The convention of *$nodes* and *$edges* are mentioned in Table 14 on page 54. The variable *$no_lonely_nodes* can be true or false and commits the framework if nodes without any edges should also be considered or not. *saveNetwork($nwName, $nwNotes, $cName, $cIP)* saves a network permanently. *getComponent($identifier)* changes the network metrics to a single component. If *$identifier* is zero or a bad value, *getComponent* sets the metrics back to the metrics of the entire network.

If the network is a complete one, the function *getPossibleEdges($nodes)* returns an array of all possible edges for the network. The *getMetric($type)* function return an array for a metric on each node. The value of *$type* can be of the following: “degree”, “betweenness”, “closeness”, “indegree”, “outdegree” and “pagerank”. *getAltersCount()* returns the amount of alter-nodes in an ego-centric network.

loadFromDB() returns *true* if the network is saved permanently.

6.1.4 Framework Usage

Here is a demonstration of the framework usage with two illustrative examples:

```
<?php
  require_once("path/to/SocioCatcher.php")

  $sc = new SocioCatcher("sparql","complete","undirected","http://dbpedia.org/sparql?format=application%2Fxml&query=");

  $nodes = array(); $nodes[0] = "rdf:type"; $nodes[1] = "dbpedia-owl:Writer";
  $edges = array(); $edges[0] = "dbpprop:influenced";

  $sc->extractNetwork($nodes,$edges);

  echo "Network identification number: " . $sc->uri . "<br>";
  echo "Amount of connected components: " . $sc->conComp . "<br>";

  for($i=0;$i < $sc->conComp; $i++) {
    $sc->getComponent($i);
    echo "Component: " . $i . "<br>";
    echo "Amount of nodes: " . $sc->nodesCount . "<br>";
    echo "Amount of edges: " . $sc->edgesCount . "<br>";
    echo "Network density: " . $sc->networkDensity . "<br>";
    print_r($sc->getMetric("degree"));
    print_r($sc->getMetric("closeness"));
    print_r($sc->getMetric("betweenness"));

    // Download button for entire network and each component
    echo '
      <form method="POST" action="path/to/pajek.php" target="">
        <input type="submit" value="Download" />
        <input type="hidden" name="uri" value="' . $sc->uri . '">
     </form><br><br>';
  }
?>
```

Listing 11: Creating a network example-code

The example in Listing 11 extracts an undirected complete network of writers with influenced relationship. Afterwards, the code outputs the network identification number and the amount of connected components. Thereafter, the for-loop puts out all metrics for the whole network and all connected components with a download button for every component. With the download buttons it is possible to download the network (component) in Pajek-Net format. In the next example we load and save a network:

```
 1: <?php
 2:   $sc = SocioCatcher("k0mycTkVgFRb6", "complete");
 3:   $sc->saveNetwork("Influenced Writer", "A simple network of influenced writer", "MZ", "localhost");
 4:
 5:   unset($sc);
 6:
 7:   $sc = SocioCatcher("k0mycTkVgFRb6");
 8:   echo "Creator name: " .   $sc->creatorName . "<br>";
 9:   echo "Network name: " .   $sc->networkName . "<br>";
10:   echo "Network notes: " .   $sc->networkNotes . "<br>";
11: ?>
```

Listing 12: Code-example for loading and saving a network

On line 2 and 3 in Listing 12 a temporary saved network is loaded and saved permanently. Line 7-10 loads a permanently saved network and outputs the name of the creator, the name of the network and the network notes.

6.1.5 Implemented Projects

As mentioned before, the framework uses two external projects, ARC2 for PHP and JUNG.

ARC2, RDF Classes for PHP

ARC2[3] is a free open source framework that runs with most server environments. It is a rewrite of ARC1 with the intention to make an easily usable, powerful framework for RDF files. It has different parsers, different serializers, with the use of MySQL a RDF storage, a SPARQL endpoint class and many other features.

The second considered PHP framework for RDF files is RAP[4] V0.9.6. We decided to use ARC2 because of its incomparably easily usable class structure and it features less errors.

JUNG Framework

JUNG[5], the Java Universal Network/Graph Framework, is an open source software library designed to model, analyze, visualize graphs. It depicts many mathematical algorithms for analyzing and preparing networks. JUNG was created by three PhD students at the University of California.

We had to resort to a Java framework because PHP does not directly support multi-threading and related efficient analysis algorithms.

6.2 Web Application

The web application merges the framework and a well designed user interface.

6.2.1 Target Group

The main intent of the web application was an applicable presentation platform for the participants of the usability study. The participants are sociology students. In addition the platform is freely accessible to all interested parties.

6.2.2 Website Structure

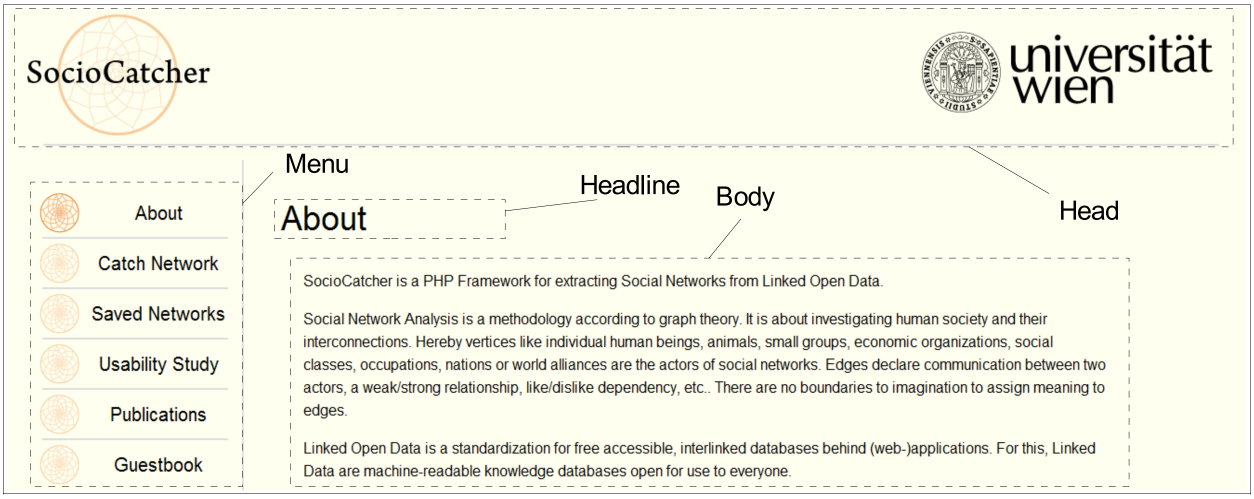
Segmentation

With the website structure, we thought of a segment for the SocioCatcher logo and the symbol of the University of Vienna with a link to the homepage of it. Besides that, the website should have a classical menu and content structure. Figure 27 shows the website layout.

[3] http://arc.semsol.org/

[4] RAP - RDF API for PHP, http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/

[5] http://jung.sourceforge.net/



SocioCatcher is a PHP Framework for extracting Social Networks from Linked Open Data.

Social Network Analysis is a methodology according to graph theory. It is about investigating human society and their interconnections. Hereby vertices like individual human beings, animals, small groups, economic organizations, social classes, occupations, nations or world alliances are the actors of social networks. Edges declare communication between two actors, a weak/strong relationship, like/dislike dependency, etc.. There are no boundaries to imagination to assign meaning to edges.

Linked Open Data is a standardization for free accessible, interlinked databases behind (web-)applications. For this, Linked Data are machine-readable knowledge databases open for use to everyone.

Figure 27: Segmentation and design of the website

Menu Structure

News: The web application normally starts with the news page. Here are the obligatory steps of the website production listed.

About: This section gives an overview on the subject of the website.

Catch Network: This menu item leads to the extraction tool.

View Network: From this section we can watch all permanently saved networks.

Usability Study: This section presents the amount of questionnaires that are filled out. If the GET variable *id* is set, the website leads to the questionnaire.

Publications: With this menu item we can download all papers depending on this web application, or at least this master thesis.

Guestbook: The guest book represents an addition for feedback or annotation on this website.

Additional Websites

Additional sections that are not within the menu structure:

Usability Study Analysis: This section lists the individual answers on the questionnaires.

Supplement: The supplement is a PDF file in addition to the usability study. [Appendix B]

6.2.3 Website Design

The website design was carried through in separated CSS (Cascading Style Sheets) files. Thus, structure and design aspects are saved in separate files. We wanted to give the website a warm, fresh and inviting touch. We thought a good base color for the website would be a pale yellow, because yellow is the only color stimulating both brain hemispheres. Additionally it is a warm color demonstrating nearness to the user [62].

Logo

As seen in Figure 28, the logo has an orange color. The color is received as fresh and bright. We chose a dreamcatcher as symbol, because, as mentioned before, SocioCatcher is a combination of the words sociomatrix and dreamcatcher.

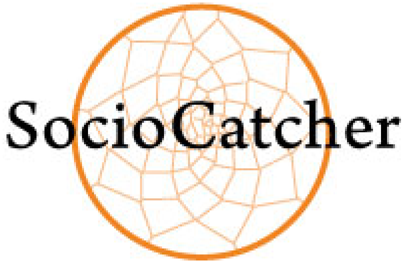


Figure 28: SocioCatcher Logo

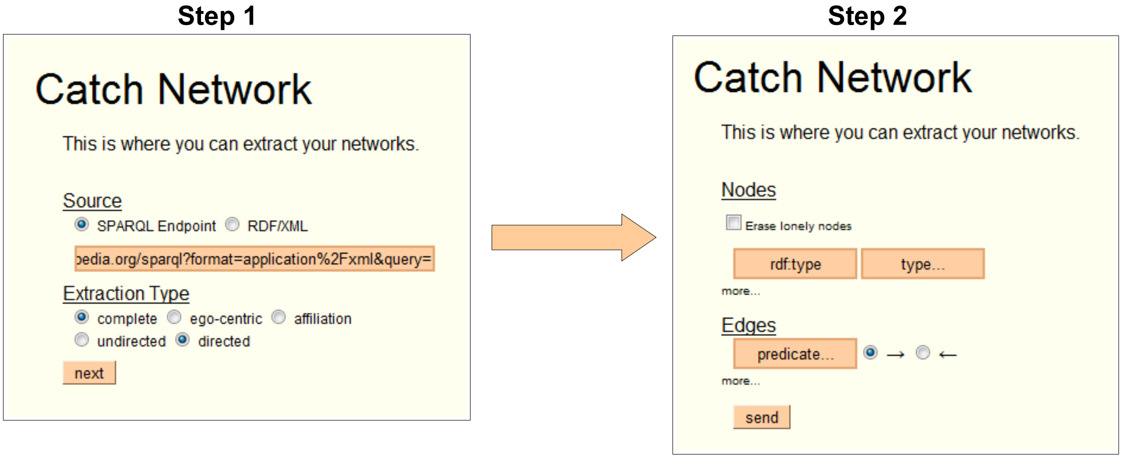
Website Impression

We wanted an austere design with the focus on functionality. We took a normal black as font color, pale yellow as background color, and an orange color for everything we wanted to attract. Orange items that are not of current interest for the user are made transparent. In addition we created mouse-over effects on the transparent items to signalize possible actions by clicking onto them.

Usability Design

We wanted to make an intuitive user interface design. As seen in Figure 29, for the network extraction we created two screens, connected with a next button.

First of all, we have to define the source of the RDF dump or SPARQL endpoint, the network type and network direction type. Secondly, we have to choose the nodes and edges, depending on our network type, with the direction of the edges and a check box for considering lonely nodes or not.



Step 1

Catch Network

This is where you can extract your networks.

Source

● SPARQL Endpoint ○ RDF/XML

ɔedia.org/sparql?format=application%2Fxml&query=

Extraction Type

● complete ○ ego-centric ○ affiliation

○ undirected ● directed

next

Step 2

Catch Network

This is where you can extract your networks.

Nodes

□ Erase lonely nodes

rdf:type | type...

more...

Edges

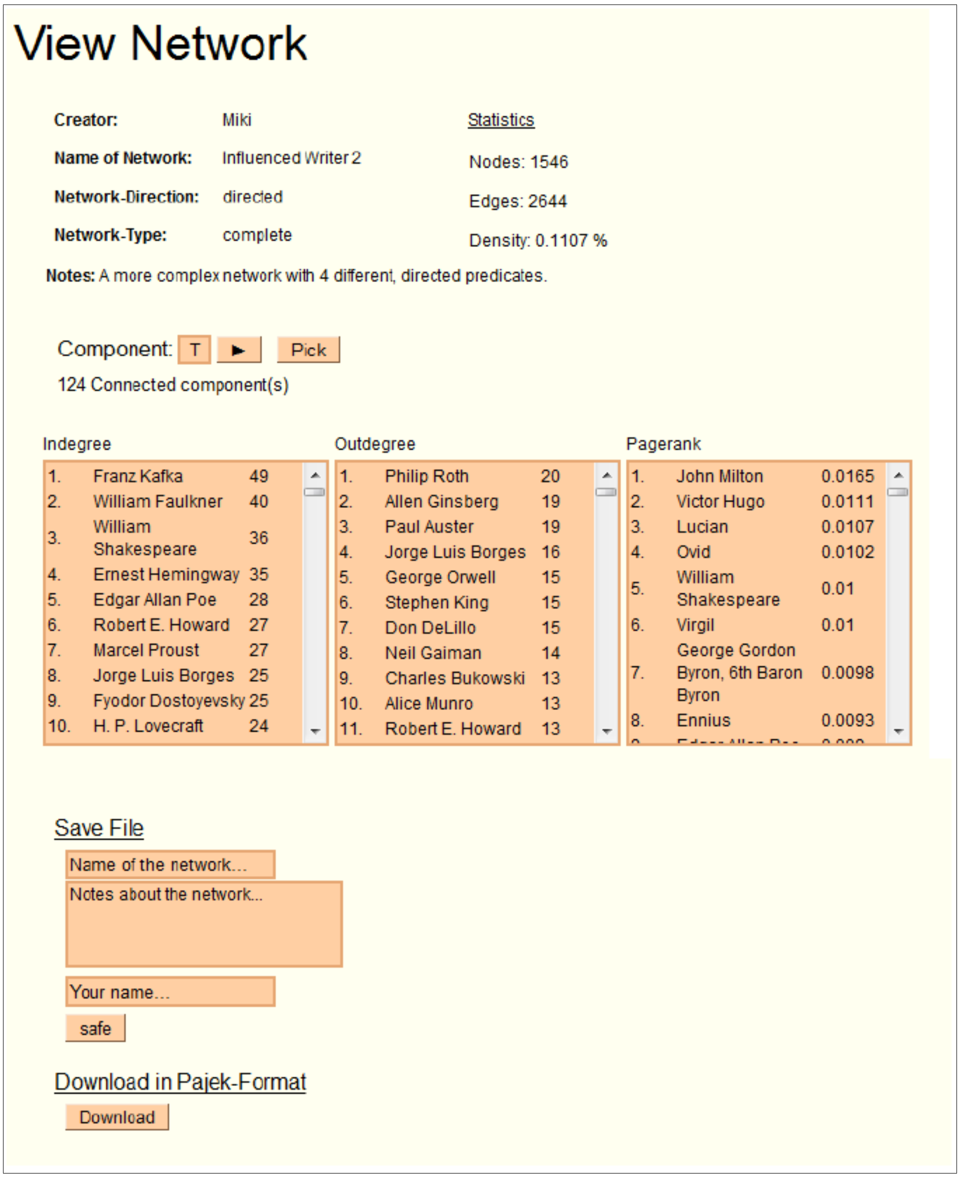
predicate... ● → ○ ←

more...

send

Figure 29: Usability design of catch network

We also thought about an extraction method with three steps. We wanted to split the second step in the selection of the nodes and then send a query to get a preselection of all possible edges. We struck to the two step method, because it takes much time to send queries to a SPARQL endpoint and the result of about 70 to 90 different possible edges would be more confusing than helping the user.



View Network

Creator: Miki

Name of Network: Influenced Writer 2

Network-Direction: directed

Network-Type: complete

Statistics

Nodes: 1546

Edges: 2644

Density: 0.1107 %

Notes: A more complex network with 4 different, directed predicates.

Component:

124 Connected component(s)

	Indegree	
1.	Franz Kafka	49
2.	William Faulkner	40
3.	William Shakespeare	36
4.	Ernest Hemingway	35
5.	Edgar Allan Poe	28
6.	Robert E. Howard	27
7.	Marcel Proust	27
8.	Jorge Luis Borges	25
9.	Fyodor Dostoyevsky	25
10.	H. P. Lovecraft	24

	Outdegree	
1.	Philip Roth	20
2.	Allen Ginsberg	19
3.	Paul Auster	19
4.	Jorge Luis Borges	16
5.	George Orwell	15
6.	Stephen King	15
7.	Don DeLillo	15
8.	Neil Gaiman	14
9.	Charles Bukowski	13
10.	Alice Munro	13
11.	Robert E. Howard	13

	Pagerank	
1.	John Milton	0.0165
2.	Victor Hugo	0.0111
3.	Lucian	0.0107
4.	Ovid	0.0102
5.	William Shakespeare	0.01
6.	Virgil	0.01
7.	George Gordon Byron, 6th Baron Byron	0.0098
8.	Ennius	0.0093

Save File

Download in Pajek-Format

Figure 30: Usability design of network analysis

With the network analysis screen, we can see in Figure 30, we parted the site into 5 parts. The first part gives an overview on the network and creator name, the network notes and some general statistics on the selected network partition or entire network. The second part is a control for switching between single components or the total network with the statistical value of how many components the network consists.

The third file is depending on the network type and network direction type. If the network type is ego-centric, there will be no statistics on the nodes. Affiliation networks and complete networks have, depending on their network direction type, three different centrality or prestige values.

The fourth part is a form for saving the network permanently. It is only visible for not-permanently saved networks.

The last part is a button for downloading the selected component of the network in Pajek format.

6.2.4 Technical Implementation

The website runs on a standard Apache 2 server with PHP 5 and MySQL service. The command line should be accessible via PHP, which is activated by default after the installation. Java should also be installed and accessible via command line.

Network Extraction

The PHP script behind the menu item “Catch Network”, *cn.php*, outputs a simple PHP generated HTML and Javascript website. The script transfers all input data as POST variables to the *vn.php* script.

Network Analysis

The *vn.php* script takes all POST or GET variables to generate a network and prepare the output data for presentation. Strictly speaking, the script distinguishes between a new network, temporary saved and permanently saved network. It also checks whether a single component is chosen or the network should be saved permanently.

The output of this script is a HTML site according to Figure 30. The script extracts the names of the nodes from its URI and sorts the statistical values to rank the nodes.

Pajek File

If we click on the download button the *pajek.php* script is opened to output the wanted pajek.net file to the user. This script only gets a network identification number and outputs the appropriate saved data on the file system to the user.

Usability Study

The usability study is saved in a simple XML file. The structure of the file is illustrated in Listing 13. An entry contains the id of the participant, all questions with q11 stands for question part 1, question 1. If a question has separate information, it is saved in c0, c1, c2, etc. elements. The questionnaire related entry elements are q11 to q16, q21 to q27 and q31 to q35.

```
<us>
   <entry>
      <q11> Value </q11>
      <q12> Value </q12>
      <q13>
         <c0> Value </c0>
         <c1> Value </c1>
         ....
         <c8> Value </c8>
      </q13>
      <q14> Value </q11>
      <q15> Value </q15>
      .....
      <datetime>YYYY-MM-DD hh:mm</datetime>
   </entry>
   <entry>
      ........
   </entry>
</us>
```

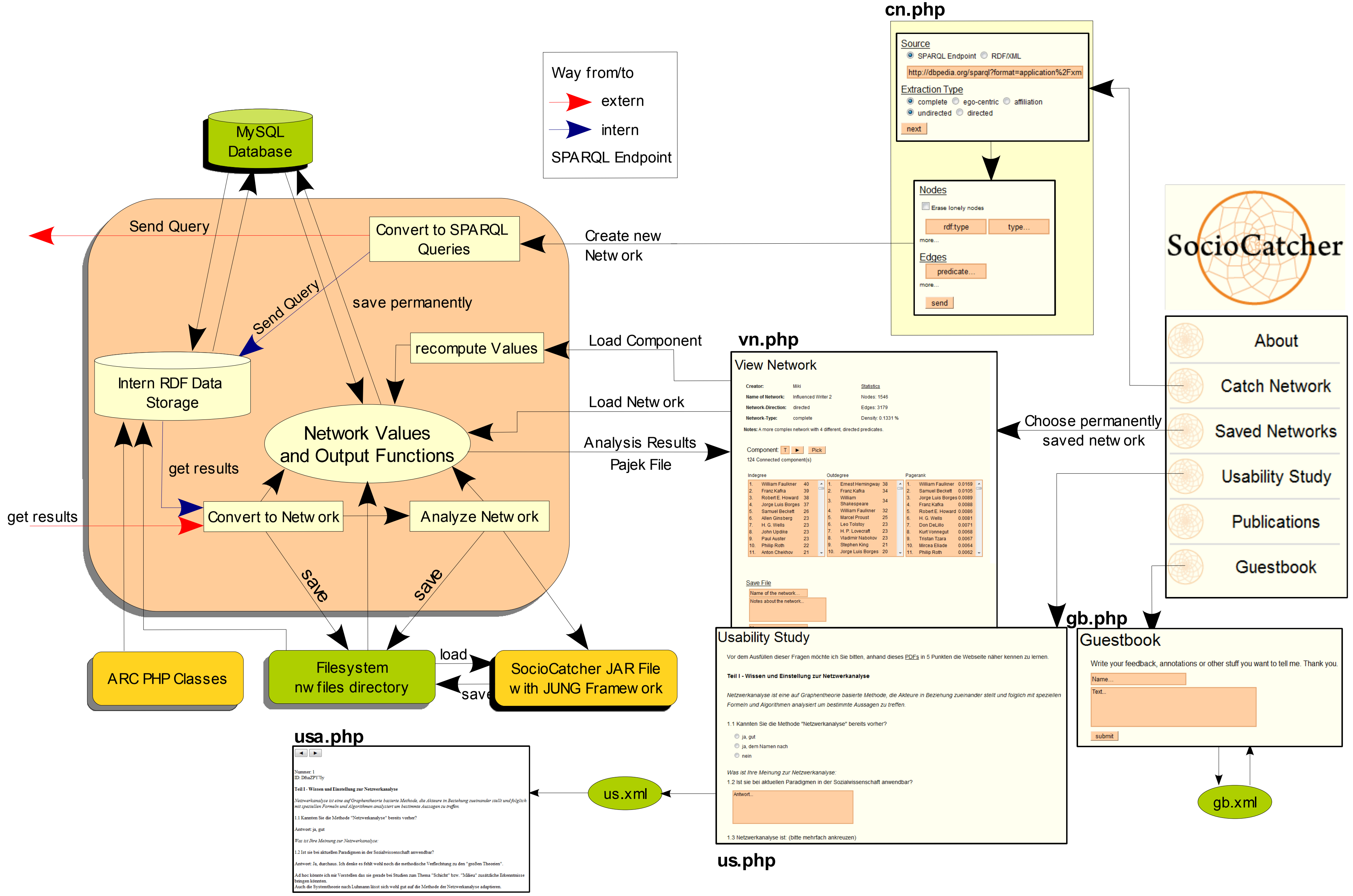
Listing 13: Example XML code for the usability study

Guestbook

The content of the guest book is also saved as XML file. Listing 14 demonstrates the XML structure.

```
<gb>
   <entry>
      <name> Value </name>
      <text> Value </text>
      <datetime>YYYY-MM-DD hh:mm</datetime>
   </entry>
   <entry>
      ........
   </entry>
</gb>
```

Listing 14: Example XML code for the guest book



View Network

Creator: Miki
Name of Network: Influenced Writer 2
Network-Direction: directed
Network-Type: complete

Statistics
Nodes: 1546
Edges: 3179
Density: 0.1331 %

Notes: A more complex network with 4 different, directed predicates.

Component: T ▶ Pick
124 Connected component(s)

Indegree			Outdegree			Pagerank		
1.	William Faulkner	40	1.	Ernest Hemingway	38	1.	William Faulkner	0.0169
2.	Franz Kafka	39	2.	Franz Kafka	34	2.	Samuel Beckett	0.0105
3.	Robert E. Howard	38	3.	William Shakespeare	34	3.	Jorge Luis Borges	0.0089
4.	Jorge Luis Borges	37	4.	William Faulkner	32	4.	Franz Kafka	0.0088
5.	Samuel Beckett	26	5.	Marcel Proust	25	5.	Robert E. Howard	0.0086
6.	Allen Ginsberg	23	6.	Leo Tolstoy	23	6.	H. G. Wells	0.0081
7.	H. G. Wells	23	7.	H. P. Lovecraft	23	7.	Don DeLillo	0.0071
8.	John Updike	23	8.	Vladimir Nabokov	23	8.	Kurt Vonnegut	0.0068
9.	Paul Auster	23	9.	Stephen King	21	9.	Tristan Tzara	0.0067
10.	Philip Roth	22	10.	Jorge Luis Borges	20	10.	Mircea Eliade	0.0064
11.	Anton Chekhov	21				11.	Philip Roth	0.0062

Save File
Name of the network...
Notes about the network...

Usability Study

Vor dem Ausfüllen dieser Fragen möchte ich Sie bitten, anhand dieses PDFs in 5 Punkten die Webseite näher kennen zu lernen.

Teil I - Wissen und Einstellung zur Netzwerkanalyse

Netzwerkanalyse ist eine auf Graphentheorie basierte Methode, die Akteure in Beziehung zueinander stellt und folglich mit speziellen Formeln und Algorithmen analysiert um bestimmte Aussagen zu treffen.

1.1 Kannten Sie die Methode "Netzwerkanalyse" bereits vorher?

○ ja, gut
○ ja, dem Namen nach
○ nein

Was ist Ihre Meinung zur Netzwerkanalyse:

1.2 Ist sie bei aktuellen Paradigmen in der Sozialwissenschaft anwendbar?

Antwort...

1.3 Netzwerkanalyse ist: (bitte mehrfach ankreuzen)

Guestbook

Write your feedback, annotations or other stuff you want to tell me. Thank you.

Name...
Text...
submit

◀ ▶

Nummer: 1
ID: DfxaZPY7Iy

Teil I - Wissen und Einstellung zur Netzwerkanalyse

Netzwerkanalyse ist eine auf Graphentheorie basierte Methode, die Akteure in Beziehung zueinander stellt und folglich mit speziellen Formeln und Algorithmen analysiert um bestimmte Aussagen zu treffen.

1.1 Kannten Sie die Methode "Netzwerkanalyse" bereits vorher?

Antwort: ja, gut

Was ist Ihre Meinung zur Netzwerkanalyse:

1.2 Ist sie bei aktuellen Paradigmen in der Sozialwissenschaft anwendbar?

Antwort: Ja, durchaus. Ich denke es fehlt wohl noch die methodische Verflechtung zu den "großen Theorien".

Ad hoc könnte ich mir Vorstellen das sie gerade bei Studien zum Thema "Schicht" bzw. "Milieu" zusätzliche Erkenntnisse bringen könnten.
Auch die Systemtheorie nach Luhmann lässt sich wohl gut auf die Methode der Netzwerkanalyse adaptieren.

7 Proof of Concept and Usability Study

This chapter deals with the answers to the questions accompanying this thesis: Is the extraction of social networks from DBpedia usable? In order to answer this question we have extracted social networks from the historical data of DBpedia and analyzed them to find patterns for making statements and conclusions on the networks.

The second question of this thesis is as follows: Is the extraction and analysis of data from DBpedia a useful approach for social science? Therefore, we set up a usability study with the goal to receive feedback from sociology students to get to know their feeling concerning this approach.

7.1 Network Analysis on DBpedia

As mentioned earlier, the network analysis consists of three steps per each network: network visualization with Pajek, statistical analysis of the network metrics and a background research on the actors. The results should be patterns and statements for the selected group of actors. These results were extracted on September the 23rd, 2010 and can differ from actual extractions.

7.1.1 Writer- Influential

The idea behind this network is to get all historical writers whom DBpedia offers, and interlink them with directed influenced attributes. Who influenced whom?

This leads to some questions:

- Which writer influenced most writers directly?
- Whose ideas represent the basis for most writers?
- Who is the most important writer of all time?

Network Extraction

The writers are identified with the predicate and object *rdf:type dbpedia-owl:Writer*.

For the arcs, DBpedia offers four different influential attributes:

- dbpedia-owl:influenced
- dbpedia-owl:influencedBy
- dbpprop:influenced
- dbpprop:influences

The attributes with *dbpprop* as namespace are raw infobox properties from Wikipedia, properties with the *dbpedia-owl* namespace are ontology properties depending on information retrieval of the Wikipedia texts. These two are not equivalent, thus we chose to take both for our analysis. *dbpedia-owl:influenced* and *dbprop:influenced* have an arc direction in one direction, the *dbpedia-owl:influencedBy* and *dbprop:influences* to the other direction.

The first intention of designing the edges would be an arc direction of how the influence “flows”. However, we have to arrange the edges as if we would ask every actor “Who influenced you?”. According to these imaginary “votes” of our actors, we arrange the edges in a direction against the influence “flow”. Thus, the starting point of our arcs should be actual living writers and the more the arcs go deeper into the network, the earlier the writers should have lived.

First of all, we tried to extract all writers with all four influenced attributes. The result amounted to 8751 writers and 2644 edges. Secondly, we tried to extract all writers that have edges with the result of 1546 nodes and 2644 edges. Thus 7205 nodes have no relationship. In our further analysis we focus only on nodes with edges.

Network Visualization

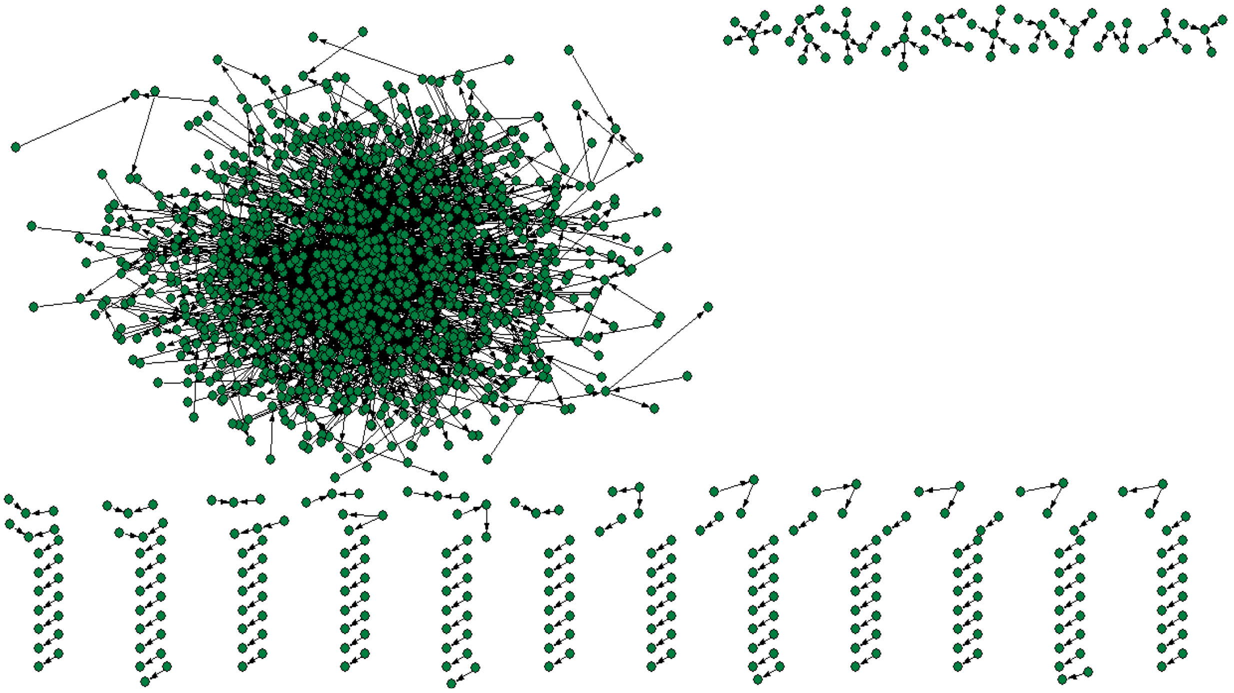


Figure 31: Visualized network of Writer influential

The visualized network (without lonely nodes) shows many very small networks and one very big one.

We explain the connected components with the focus on different regions and literature genres. The big network consists of the temporary mainstream, while the small ones embrace regional writers like Chinese or Hungarian ones (e.g. the network of Attila József, Mihály Babits, Gyula Juhász, Gyula Illyés, László Németh and János Pilinszky) or specific genres, such as genres of science fiction and fantasy (e.g. the network of C.J. Cherryh, Andre Norton, Marion Zimmer Bradley, Linnea Sinclaire and Mercedes Lackey).

Statistical Analysis

The complete network consists of 7329 connected components, with 8751 nodes and 2644 edges and shows a network density of 0.0035%, which is very low. Without the lonely nodes the networks has 1546 nodes and 2644 edges with a network density of 0.1107%.

The connected components are:

Amount	Nodes	Edges
1	1252	2473
3	6	5
3	5	4
5	4	3
17	3	2
95	2	1
7205	1	0

Table 17: Connected components of influenced writer

The indegree allocation (without lonely nodes) of our actors shows a steep curve. 627 nodes have no incoming edge, 260 nodes have one incoming edge, up to one node with an indegree of 49.

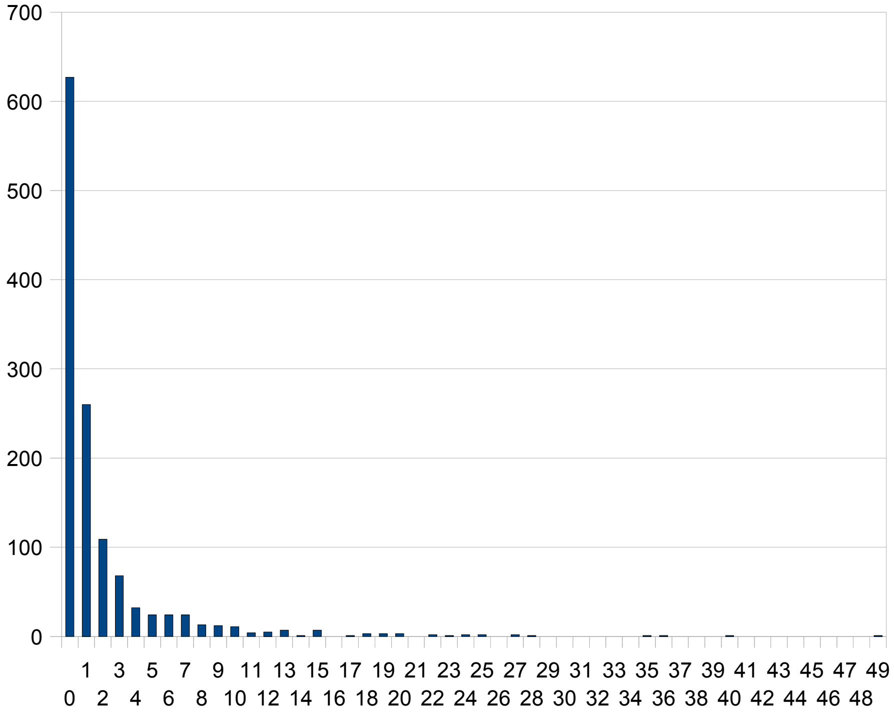


Figure 32: Indegree prestige allocation

The more indegree prestige a person has, the more a person influences people directly. With these metric, we can say, the most famous writer of his time and thereafter is Franz Kafka[1], a famous novelist at the beginning of the 20th century, with 49 writers whom he influenced directly. The next in line are William Falkner[2], a Nobel-prize winning American novelist in the 20th century, William Shakespeare[3], a famous poet and playwright, living from the 16th to the 17th century, and Ernest Hemingway[4], an American author and journalist, with 40, 36 and 35 directly influenced persons. Table 18 depicts the 20 most important writers depending on the indegree prestige value.

Rank	Writer	Indegree
1.	Franz Kafka	49
2.	William Faulkner	40
3.	William Shakespeare	36
4.	Ernest Hemingway	35
5.	Edgar Allan Poe	28
6.	Robert E. Howard	27
6.	Marcel Proust	27
8.	Jorge Luis Borges	25
8.	Fyodor Dostoyevsky	25
10.	H. P. Lovecraft	24
10.	H. G. Wells	24
12.	Leo Tolstoy	23
13.	Anton Chekhov	22
13.	Gustave Flaubert	22
15.	Samuel Beckett	20
15.	Stephen King	20
15.	John Milton	20
18.	Vladimir Nabokov	19
18.	G. K. Chesterton	19
18.	Mark Twain	19

Table 18: 20 most important writers depending on indegree prestige

With the help of Pajek, it is easy to compute proximity prestige, as mentioned in Section 3.2.3 on page 24. Proximity prestige on a node regards the surrounding of the node. This metric calculates every node that reaches it. Due to this method, we can get a closer look on how close an actor is to others that can reach him, and can get a clue on their importance. With this metric, the most centered node is represented by William Shakespeare. He is followed by Ovid[5], or Publius Ovidius Naso, who was a Roman poet between 43 BC and 17 or 18 AD. The third ranked is Edmund Spenser[6], an English poet of the 16th century. The fourth is John Milton[7], an English poet and author of the early 17th century, famous for his epic poem, Paradise Lost, followed by Victor-Marie Hugo[8], a French poet, playwright, novelist, essayist, visual artist, etc. of the 19th century.

[1] http://dbpedia.org/resource/Franz_Kafka

[2] http://dbpedia.org/resource/William_Falkner

[3] http://dbpedia.org/resource/William_Shakespeare

[4] http://dbpedia.org/resource/Ernest_Hemingway

[5] http://dbpedia.org/resource/Ovid

[6] http://dbpedia.org/resource/Edmund_Spenser

[7] http://dbpedia.org/resource/John_Milton

[8] http://dbpedia.org/resource/Victor_Hugo

Rank	Writer	Proximity
1.	William Shakespeare	0.136
2.	Ovid	0.129
3.	Edmund Spenser	0.126
4.	John Milton	0.126
5.	Victor Hugo	0.123
6.	Dante Alighiery	0.119
7.	Virgil	0.118
8.	Lucian	0.114
9.	Geoffry Chaucer	0.112
10.	Cicero	0.111
11.	François-René de Chateaubriand	0.111
12.	Petrarch	0.110
13.	Gustave Flaubert	0.109
14.	Alexander Pushkin	0.107
15.	Edgar Allan Poe	0.107
16.	Christopher Marlowe	0.106
17.	Walter Scott	0.104
18.	Torquato Tasso	0.101
19.	Miguel de Cervantes	0.100
20.	Alphonse de Larmatine	0.990

Table 19: 20 most important writers depending on proximity prestige

The next factors we would like to analyze is which intellectual heritage accounts for our writers until today. We can analyze this by applying the factor of rank prestige, in our case page rank. The writers are significant earlier in time-line, than our indegree ranking. The first one in place is John Milton, followed by Victor-Marie Hugo. The third ranked is Lucian of Samosata[9], an Assyrian rhetorician who wrote in Greek language in the 2nd century. In the fourth place is Ovid, before William Shakespeare.

These actors are the most influencing people through out time, according to rank prestige:

Rank	Writer	Page Rank
1.	John Milton	1.65 %
2.	Victor Hugo	1.11 %
3.	Lucian	1.07 %
4.	Ovid	1.02 %
5.	William Shakespeare	1.00 %
6.	Virgil	1.00 %
7.	George Gordon Byron	0.98 %
8.	Ennius	0.93 %
9.	Edgar Allan Poe	0.90 %
10.	Dante Alighieri	0.87 %
11.	Johann Wolfgang von Goethe	0.82 %
12.	Alexander Pushkin	0.80 %
13.	Daniel Defoe	0.79 %
14.	Cicero	0.76 %
15.	Jonathan Swift	0.74 %
16.	Geoffrey Chaucer	0.69 %
17.	Petrarch	0.67 %
18.	Gustave Flaubert	0.66 %
19.	H. G. Wells	0.64 %
20.	Franz Kafka	0.63 %

Table 20: 20 most important writers depending on page rank

[9] http://dbpedia.org/resource/Lucian

We created a new chart, showing all writers that are in the top 20 of proximity prestige and rank prestige, treating both values equally. We get thirteen actors which are, nominally, the most important writers in history:

Rank	Writer	Proximity	Page Rank	Indegree
1.	John Milton	4.	1.	15.
2.	William Shakespeare	1.	5.	3.
3.	Ovid	2.	4.	133.
4.	Victor Hugo	5.	2.	33.
5.	Lucian	8.	3.	72.
6.	Virgil	7.	6.	72.
7.	Dante Alighieri	6.	10.	32.
8.	Edgar Allan Poe	15.	9.	5.
9.	Alexander Pushkin	14.	12.	33.
10.	Cicero	10.	14.	157.
11.	Geoffrey Chaucer	9.	16.	133.
12.	Petrarch	12.	17.	133.
13.	Gustave Flaubert	13.	18.	13.

Table 21: Thirteen most important writer

7.1.2 Soccer-players on the same Teams

The next network we would like to analyze is a network of soccer players. The career of a soccer player is built up of many stations. We would like to analyze the transfer behavior of the soccer player. Are there any patterns of player movement? What are the common properties of network components?

Network Extraction

We extract an affiliation network with all *rdf:type dbpedia-owl:SoccerPlayer* as nodes. The edges are extracted through the container attribute behind the property *dbpprop:clubs*. Therefore, in consideration of a threshold of three, all players that played for same three clubs, not imperative at the same time, are interconnected.

We chose three as threshold, because one mutual club is common, two mutual clubs are uncommon, even though possible, three mutual clubs we find conspicuous and see a pattern.

We also restrict our analysis to nodes that have an interconnection.

Network Visualization

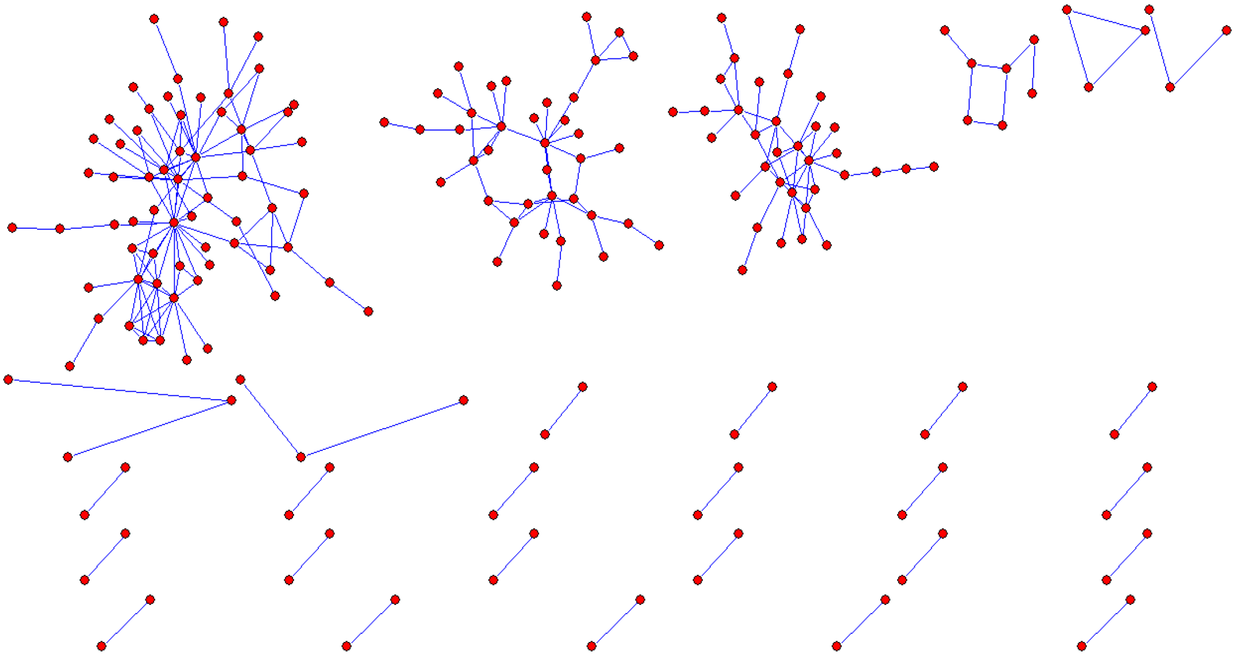


Figure 33: Visualized network of soccer players at the same three teams

The visualized network shows three major components, one component of 7 nodes and 25 small components.

Results

Component with 63 nodes and 103 edges:

The biggest component exhibits a network density of 5 %. The middle of this component is the player Cornell Glen, with the highest degree, closeness and betweenness centrality. The second on in line is John Wolyniec, also having the second best scores in all three metrics.

We found all players are playing in the Major League Soccer in the USA. The composition of this component is no surprise, because of their unusual league system. The teams in this league get mixed up among each other very often. The players change their team within the league nearly every season. As conclusion, our biggest network component focuses on Major League Soccer players.

Component with 38 nodes and 46 edges:

This is the second-biggest component and has with 7 % a higher density than our component before. The most important actor in this component is Craig Bellamy, with the highest rank in all three centrality metrics (degree, closeness, betweenness). After careful analysis of the nodes, we also saw a common pattern in this component. The common teams of this network are all playing in the English Premier League. Most of the players are of English nationality, but we also find a German, a Norwegian, a Bermudian, etc. playing in the Premier League. This network is a network of Premier League players.

Component with 34 nodes and 47 edges:

This is our third major component which with 8 % shows an even higher network density than the ones before. This network has also a key player who is highest ranked within all three centrality metrics, Christian Vieri. To find the common attributes here, we analyzed the actors with the result that this network mainly consists of players that played in the Italian Serie A, but some players played also in the Spanish La Liga and the Portuguese league. We found out that especially Brazilian players, like Ronaldo, Rivaldo, Roberto Carlos, Edmundo, etc. have played in this leagues and interconnected them. Nevertheless the main part of the network is of players in the Italian league.

Other components:

The component with 7 nodes has a very high network density of 33%. This component is also a network of soccer players in the English Premier League. The four components with three edges are also allocated to the Premier League. Among the components that connect, only two nodes have miscellaneously common attributes, like a network of the French or Danish league.

As a conclusion on this entire network we can say, that most players change club within the Major League Soccer, the Premier League and also within the Mediterranean Leagues (Italy, Spain, Portugal). With exception of the Mediterranean Leagues we haven't found any international transfer patterns. With exception of the Major League Soccer, this are nominally the best leagues in the world. It is conspicuous that there are no networks of other leagues, maybe there is no data available on them.

7.1.3 Scientist Advisers

The next network on DBpedia data deals with important historical scientists, their doctoral advisers and academic advisers and their influence. So, the main question concerning this network is, which scientist is the father of all scientists. Who is the most important scientist in history?

Network Extraction

Scientists in DBpedia data are identified via the *rdf:type dbpedia-owl:Scientist*. For the interconnection there are six attributes:

- dbpedia-owl:academicAdvisor
- dbpedia-owl:doctoralStudent
- dbpedia-owl:doctoralAdvisor
- dbpedia-owl:notableStudent
- dbpedia-owl:influenced
- dbpedia-owl:influencedBy

We arrange the arc-direction like the arcs in Section 7.1.1, so that it directs from the students to the advisers or from the scientists being influenced to the scientists who have an impact themselves.

Network Visualization

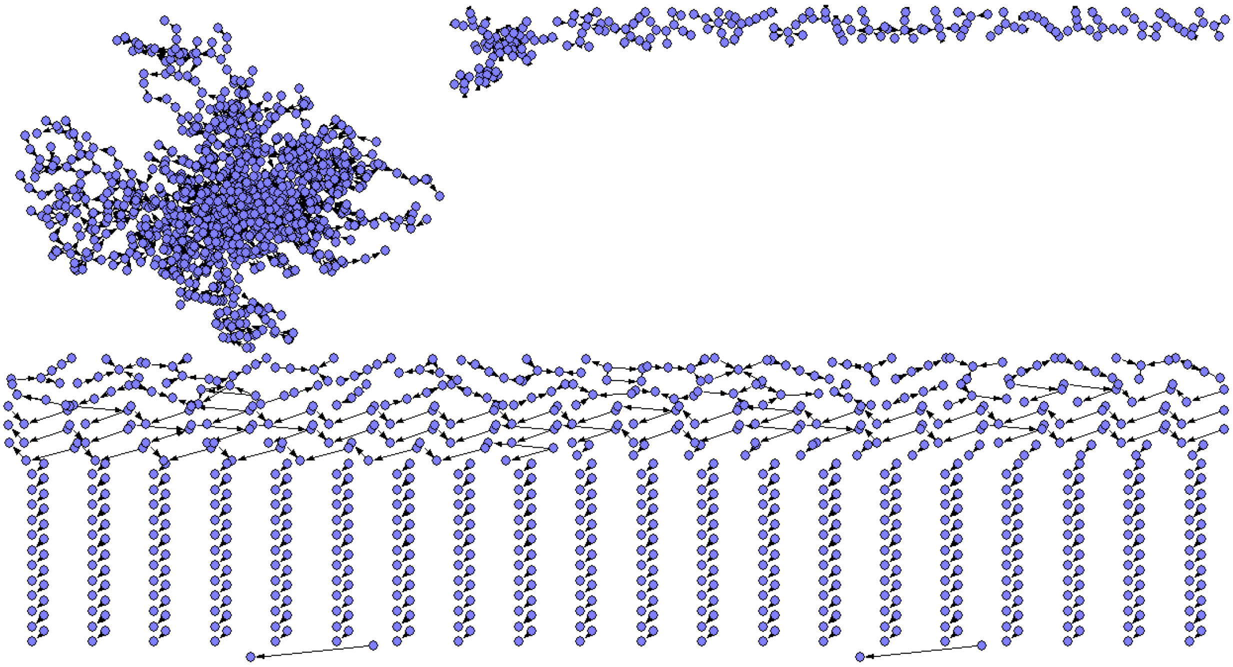


Figure 34: Visualized network of scientists

The network is very similar to the network of writers in Section 7.1.1. It consists of one major component and many small components. The major components will be of mainstream science in the USA, Europe an Japan, while the small components will refer to regional scientists and side-topics in research.

Results

The network consists of 8475 (2060 connected) nodes and 1842 edges with a very low network density of 0.00002% (0.043%).

Amount	Nodes	Edges
1	1044	1174
1	57	59
1	12	12
1	11	10
1	10	9
1	9	8
3	8	7
2	7	6
351	[2,6]	
6415	1	0

Table 22: Connected components of scientist network

The indegree allocation is illustrated in Figure 35. It is obvious that the allocation is a curve. About half of the nodes have an indegree of zero, followed by 700 with indegree 1. After that, the curve falls to 200 nodes with an indegree of two, and only about 50 nodes have indegree 3, and so on.

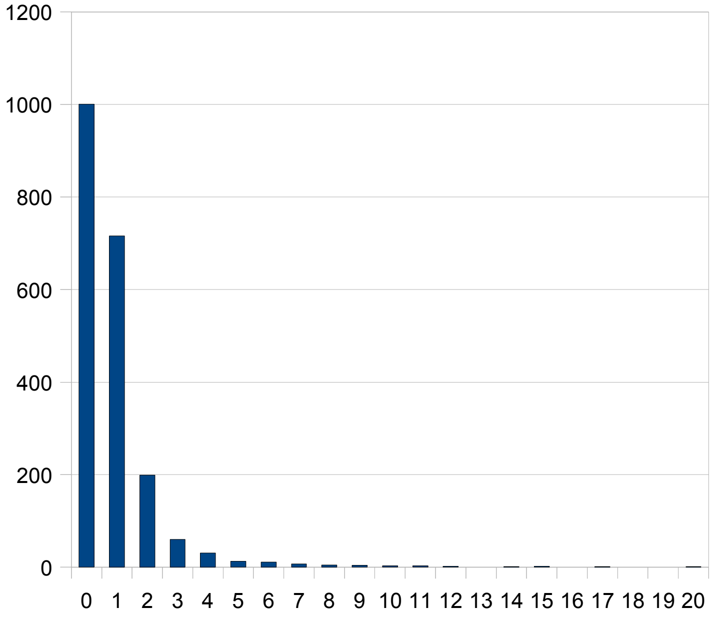


Figure 35: Allocation of indegree prestige

The top 20 scientists with the highest indegree prestige are:

Rank	Scientist	Indegree	Rank	Scientist	Indegree
1.	Ernest Rutherford	20	11.	Hermann Emil Fischer	10
2.	Robert Bunsen	17	11.	Felix Klein	10
3.	Charles Darwin	15	11.	August Wilhelm von Hofmann	10
3.	Justus von Liebig	15	14.	Karl Weierstrass	9
5.	J. J. Thomson	14	14.	John Archibald Wheeler	9
6.	Adolf von Baeyer	12	14.	Werner Heisenberg	9
6.	Arnold Sommerfeld	12	14.	Sigmund Freud	9
8.	Max Planck	11	18.	Albert Einstein	8
8.	Walther Nernst	11	18.	Isaac Newton	8
8.	Enrico Fermi	11	18.	David Hilbert	8

Table 23: Top 20 indegree prestige of scientist network

Table 23 illustrates the most important scientists that directly advised or influenced others. The most important person for his surrounding was Ernest Rutherford[10], the father of nuclear physics. The second one in place is Robert Bunsen[11], a German chemist. The third place is shared by Charles Darwin[12], an English naturalist, and Justus von Liebig[13], also a German chemist.

[10] http://dbpedia.org/resource/Ernest_Rutherford

[11] http://dbpedia.org/resource/Robert_Bunsen

[12] http://dbpedia.org/resource/Charles_Darwin

[13] http://dbpedia.org/resource/Justus_von_Liebig

Table 24 gives a more detailed impression on the surroundings of an actor. Here we have Johann Friedrich Gmelin[14] and his father Philipp Friedrich Gmelin[15], both naturalists, in the proximity ranking on place 1 and 3. The second one in place is Friedrich Stromeyer[16], a German Chemist.

Rank	Scientist	Proximity
1.	Johann Friedrich Gmelin	0.0226
2.	Friedrich Stromeyer	0.0225
3.	Philipp Friedrich Gmelin	0.0198
4.	Louis Nicolas Vaquelin	0.0196
5.	Felix Klein	0.0183
6.	Carl Friedrich Gauss	0.0182
7.	J.J. Thomson	0.0181
8.	Burchard Mauchart	0.0177
9.	Antoine François	0.0173
10.	Adam Sedgwick	0.0171
11.	Ernest Rutherford	0.0169
12.	Justus von Liebig	0.0164
13.	Elias Rudolph Camerarius Jr.	0.0160
14.	Johann Friedrich Pfaff	0.0159
15.	Abraham Gotthelf Kästner	0.0159
16.	Ferdinand von Lindemann	0.0157
17.	Thomas Jones	0.0157
18.	John Strutt	0.0154
19.	Julius Plücker	0.0154
19.	Rudolf Lipschitz	0.0154

Table 24: Top 20 proximity prestige of scientist network

The ranking, which intellectual heritage accounts for our scientists until today, offers the opportunity for another issue for our investigation. It gives a clue on which scientist is the most important. In this ranking, Friedrich Stromeyer takes the lead before Abraham Gotthelf Kästner[17], a German mathematician. Third is Christian August Hausen[18], mathematician, astronomer and physician. Fourth is Felix Klein[19], also a German mathematician. With Carl Friedrich Gauss[20] there is even a fifth German and fourth mathematician within the top 5 of this ranking.

Rank	Scientist	Page Rank
1.	Friedrich Stromeyer	1.16 %
2.	Abraham Gotthelf Kästner	1.07 %
3.	Christian August Hausen	0.98 %
4.	Felix Klein	0.95 %
5.	Carl Friedrich Gauss	0.94 %
6.	Johann Friedrich Pfaff	0.92 %
7.	Ernest Rutherford	0.90 %
8.	Johann Christoph Wichmannshausen	0.85 %
9.	Otto Mencke	0.73 %
10.	Johann Friedrich Gmelin	0.73 %
11.	Ferdinand von Lindemann	0.70 %
12.	Robert Bunsen	0.68 %
13.	Philipp Friedrich Gmelin	0.66 %
14.	Justus von Liebig	0.66 %
15.	Burchard Mauchart	0.65 %
16.	J. J. Thomson	0.62 %
17.	John Strutt	0.62 %
18.	Karl Wilhelm Gottlob Kastner	0.61 %
19.	Christoph Mangold	0.59 %
20.	Elias Rudolph Camerarius Jr.	0.59 %

Table 25: Top 20 rank prestige of scientist network

[14] http://dbpedia.org/resource/Johann_Friedrich_Gmelin

[15] http://dbpedia.org/resource/Philipp_Friedrich_Gmelin

[16] http://dbpedia.org/resource/Friedrich_Stromeyer

[17] http://dbpedia.org/resource/Abraham_Gotthelf_K%C3%A4stner

[18] http://dbpedia.org/resource/Christian_August_Hausen

[19] http://dbpedia.org/resource/Felix_Klein

[20] http://dbpedia.org/resource/Carl_Friedrich_Gauss

For our final ranking, we treat proximity and rank prestige equal and compute a ranking, with indegree as decision base on an equal result. Table 26 illustrates the scientists that find themselves in the top 20 of proximity and rank prestige. The most important scientist in this new ranking is Friedrich Stromeyer, followed by Felix Klein and Carl Gauss. Son and Father Gmelin are in the fourth and fifth place.

It is conspicuous, that most of the scientists in this list are German. With Ernest Rutherford, the first British scientist is ranked in place 7. This ranking nominally shows that German scientists are most important in scientific history.

Rank	Scientist	Proximity	Page Rank	Indegree
1.	Friedrich Stromeyer	2.	1.	85.
2.	Felix Klein	5.	4.	11.
3.	Carl Friedrich Gauss	6.	5.	18.
4.	Johann Friedrich Gmelin	1.	10.	145.
5.	Philipp Friedrich Gmelin	3.	13.	334.
6.	Abraham Gotthelf Kästner	15.	2.	145.
7.	Ernest Rutherford	11.	7.	1.
8.	Johann Friedrich Pfaff	14.	6.	334.
9.	J. J. Thomson	7.	16.	5.
10.	Burchard Mauchart	8.	15.	334.
11.	Justus von Liebig	12.	14.	3.
12.	Ferdinand von Lindemann	16.	11.	54.
13.	Christian August Hausen	24.	3.	334.
14.	Elias Rudolph Camerarius Jr.	13.	20.	334.
15.	John Strutt	17.	17.	85.

Table 26: Fifteen most important scientists

7.1.4 Architect Teams

The last network we deal with is a network of architects. Our leading question is: How many architects are working in teams and which size do these teams have?

Network Extraction

For this network we use *rdf:type dbpedia-owl:Architect* for the definition of the nodes. We receive the edges through container that are defined through the attributes *dbpedia-owl:significantProject* and *dbpedia-owl:significantBuilding*.

Network Visualization

This network has no complex parts, therefore, the analysis has no need for application of advanced metrics.

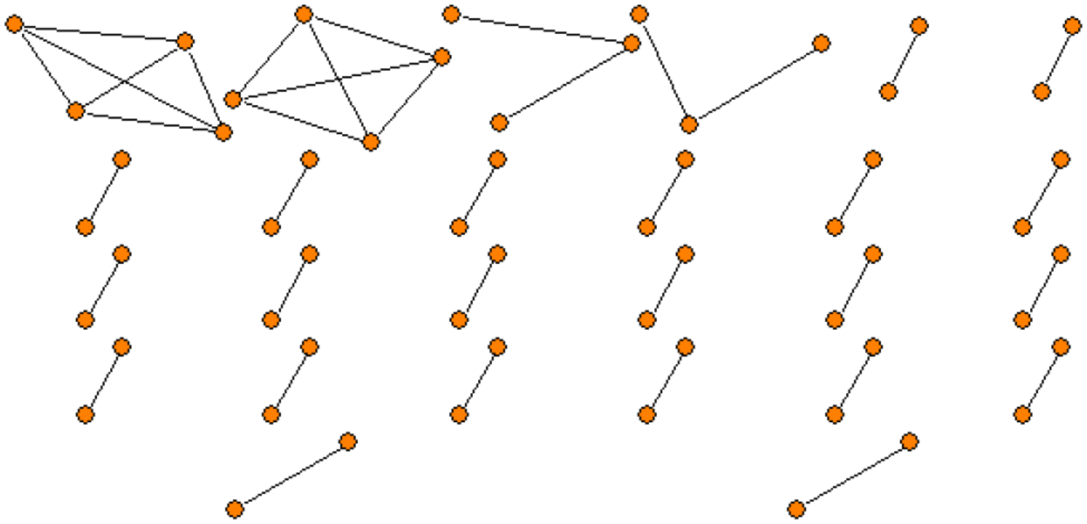


Figure 36: Network visualization of architect teams

Results

Within this network we have 26 components with more than one node. Some individual components are incorrect and some individual edges as well. For instance the node *List_of_Gaudí_buildings* is connected with *Antoni Gaudí*. Another discovery was a container property of only the country Belgium or the city of Vienna which falsely interconnected two nodes. So we have 24 real architect teams and 602 single architects in our network. Typically each component shows a density of 100%. Table 27 illustrates the buildings where the architect teams worked together. Every team only worked on one project.

Amount	Team of	Created Building
2	4	New England Biolabs, Palace of Soviets
22	2	Wembley Stadium, Seattle Central Library, Guthrie Theater, US Capitol, Royal Palace of Belgium, Iguada Cemetery, Eaton Hall, Centre Pompidou, National Museum of Finland, Hopetown House, Los Angeles City Hall, Ohio State House, Huntington Library, Moscow State University, Bath Abbey, Pearl River Tower, Burj Khalifa, PacBell Building, World's Columbian Exposition, Commonwealth War Graves Commission, Kemp Town, Reston Virginia

Table 27: Architect teams by buildings

7.2 Usability Study

In order to assure that the chosen approach is useful, we undertook a small usability study. This study reveals the attitude of sociology students to the social network analysis in general and helps to get knowledge on their opinion and previous knowledge. In addition, we asked questions concerning the previous knowledge of DBpedia, the general technical ability, and their opinion on the web and social software (in context to social networks). The third questionnaire part consists of questions on the technical and conceptual part of the master thesis. For the questions in detail see Section 5.4.3 on page 50.

7.2.1 Accomplishment

For the accomplishment we used a pyramid scheme. Starting with the interview partner we had an expert interview at the start of this thesis [Appentix A]. We sent an e-mail template with text and a number of codes for the questionnaire identification-numbers. The interview partner forwarded the e-mail to another 25 sociology students. The callback on this approach brought 9 filled in questionnaires. Considering our intention of receiving 7 filled in questionnaires, this proves a success.

7.2.2 Participants

We asked many questions in order to get to know the specific background of the participants. We wanted to classify them into categories to allow a better rating of their answers. The factors were previous knowledge and personal attitude on the main topics. Even if the previous knowledge was very low, an additional document gave the basics on both main topics, social network analysis and Linked Data [Appendix B].

7.2.3 Analysis

Previous Knowledge on Social Network Analysis

The previous knowledge of the participants on social network analysis is vague. Two people said they know the social network analysis very well, five said they at least know the name of this method and two said they had not heard of the social network analysis until receiving the questionnaire.

The two participants who know the social network analysis very well, also share the opinion that there will be a focus on this method in the future. The other participants point out its complexity, but nevertheless find it interesting. Others note that outside of sociology this method is already in use. Five people think that the method is only a side topic in social science, but moreover the majority thinks that they will be more often used in the future. Concerning the question whether social network analysis is convenient to answer actual questions of the social science, all participants agree.

The question on computational social science is also very clear for all protagonists. The majority is for a change to computational social science and distinguish that quantitative methods nowadays are not possible without the use of specific programs (for example SPSS).

At the question, what they think on interdisciplinary work (e.g. computer science and sociology) all participants agree. As a conclusion, even if the previous knowledge was not very high, the overall opinion on the social network analysis is absolutely positive.

Previous Technical Knowledge

We asked the participants, how they think about their technical abilities and how they would rate them on a scale between 1 and 10. The results showed an average result of 6. With the questions on social software, privacy and research fields on the web, the opinions are different. Four participants think that there are enough research fields on the web, four proposed input for more interesting topics, one gave no statement. Four people use social software and are very privacy affine, five people do not use social software at all. To the question, whether they had previous knowledge on DBpedia, all except one said they had none. One participant said he or she at least knows the name. Nevertheless, four people thought that DBpedia is a good source for extracting social networks, three had no opinion and one gave no statement.

As a conclusion, the majority has an average technical knowledge and does not know DBpedia. Half of the people is open minded about DBpedia and the internet, the other half is not very interested in internet-related topics.

Usability

Seven participants think that automatic data extraction from DBpedia or other Linked Data sources is a good approach, two think it is a moderate approach. Five people think the website for the extraction of social networks from DBpedia is useable for research with the aid of more powerful programs. four people share the opinion that the raw data of DBpedia has to be eyed critically according to its correctness, so and thus it is not suitable for research.

The extraction tool of this website is sufficient for six participants, one thinks it is well chosen and two think it is inadequate. The people who think it is inadequate would have liked more help for easier use, because they found it very technical. The analysis tool is also sufficient for most people, one thinks it is well chosen, and one thinks that it is inadequate. The criticism of this person is, that with this part of the website, there is too less aid for users who do not know about the methodology of social network analysis at all.

In general, the participants pointed out the need of improvements about this website, that more help on the processing steps and data input would be important. More information on DBpedia would also help. Network visualization or even a better inclusion of Pajek should be improved, if possible. The data source should be displayed on the analysis window. Also better malfunction messages would be helpful.

A deeper criticism on the approach is the scrupulosity on the automatic extraction of relationships. These relations are a sensible thing. To abstract and generalize such relationships via automatically processed algorithms is a problematic thing to do. Also the standing of the individual is a factor to be considered.

Two people also think that Wikipedia is not accepted in scientific communities and therefore, raw data obtained out of Wikipedia is questionable for scientific research.

7.3 Conclusion

We saw respectable results in this chapter. In Section 7.1 we saw four networks extracted and analyzed with the SocioCatcher framework and a little help of Pajek. This analysis tells us that the extraction and analysis is useable, and produces applicable results. We also saw that the data is not free of errors and, for a very sensitive analysis, the data should also be reviewed and corrected.

There is also a general question concerning the raw data. We have no evidence that the data is complete. With the network of soccer player in Section 7.1.2 we saw three major networks, one of the Major League Soccer, one of the Premier League an one of the Mediterranean Leagues. The French league as one of the top 5 leagues in Europe was not contained in the network. This holds also other leagues. Therefore, we have to choose the networks and handle the results and the ensuring knowledge with care.

The usability study was a success in most cases. We wanted to know whether students of sociology think that this method is applicable with DBpedia. As a surprise we found out that only a few participants know about the methodology of the social network analysis and even less know about DBpedia. Nevertheless the overall consensus was positive. The two participants who had known about the social network analysis method before, gave a complete positive feedback.

Some students proposed user interface improvements. The majority wanted more information on DBpedia and the social network analysis, which was not the intention of this tool (the tool was created for people with knowledge on both topics). No one questioned the approach itself.

The main critique was on the raw data of DBpedia. Is Wikipedia (and with that DBpedia) not useable for social research, because such data is not accepted in research? Five questioned people, including the two people with good knowledge on the social network analysis, think that it is capable, four think it is not, which would be a interesting controversial issue. According to the two sociology students who know the social network analysis well, we can assume that it is also applicable for social research.

Finally we can say, that social network extraction and analysis from DBpedia data is useable, which we saw in our analysis of the four networks. According to the two sociology students who have knowledge in social network analysis it is useful too.

Summary and Future Work

Within this thesis we combined two emerging research areas, the social network analysis and Linked Data. Strictly speaking, we investigated the usability and usefulness of extracting and analyzing social networks out of DBpedia data. We created the SocioCatcher framework for extracting social networks from single Linked Data sources via their SPARQL endpoint, and the necessary infrastructure for analyzing these networks by using common network measures. Additionally we created a web application with a simple user interface on-top of this framework. We extracted and analyzed four different networks and got remarkable results:

We extracted a network of writers and learned, that the most important writer in history is John Milton, the author of Paradise Lost. The most directly influencing writer was Franz Kafka and the writer with the largest fame (according to proximity prestige) was William Shakespeare.

We used the same technique on a network of scientists with a similar success. The most important scientist of his time, who influenced the biggest amount of scientists, was Ernest Rutherford. Except for him, the most important scientists in history were mostly German. The best proximity prestige value, and thus the largest fame, scored Johann Friedrich Gmelin, a German medic and botanist. The scientist whose intellectual heritage influenced most scientists was Friedrich Stromeyer, a German chemist.

For the two other networks we applied an indirect extraction method. We extracted a network of football players, who we defined as interlinked, if they were playing in the same clubs three times. As a result we received connected components with similarities on different national leagues. We received a network of the Major League Soccer, a network of the Premier League, and, even more interesting, a network of the Mediterranean leagues (Spain, Portugal and Italy). For these networks it can be concluded, that while soccer players in the United States and England stay in their leagues, the players in the Spanish, Portuguese and Italian league transfer more often between these leagues. We also found out that many leagues are absent in the data.

The last network we extracted was about architects, working on buildings. We wanted to know whether architects work in groups or alone. This network was not complex, so it gave us the opportunity to inspect the extracted data in detail. We even found some errors in the raw data.

Especially the soccer player and the architect networks gave us insight into the quality of the underlying raw data. In the case of the architect network, we had to control nodes and edges manually. In case of the soccer player network, we may even think about another extraction method for absent raw data.

Additionally, we wanted to get an expert opinion from sociology students on our approach and made a usability study. We found out that social network analysis as a scientific method is not commonly known among sociology students. From nine participants, only two knew about this method in more detail. These two students and some others confirmed the approach completely. Others shared a critical opinion about this approach, especially on the quality of raw DBpedia data. Two participants even said that they were not sure whether or not data from Wikipedia (and with that DBpedia) is acceptable for serious research. This is a really interesting outcome, whose answer is out of the scope of this thesis.

Given our extracted networks and the expert feedback we regard our approach as success. DBpedia can be used to extract and analyze social networks, even for research in social science.

We received a lot of positive feedback on our extraction and analysis tool, which we could improve in terms of user interface design and usability. This tool can help social science students and scientists without any technical knowledge to extract networks, and on top of this, the tool can advance even the social network analysis method to make it more popular. If this is going to happen, the Linked Data community will also share an advancing popularity and receive a new focus for their data.

In this thesis only DBpedia was analyzed. There were many other Linked Data data sources that can be analyzed with the same framework.

References

[1] T.B. Bottomore, Maximilien Rubel, *Karl Marx - Selected Writings in Sociology & Social Philosophy*, 1st ed. McGraw-Hill Humanities/Social Sciences/Languages, 1964.

[2] R.L. Breiger, “The Analysis of Social Networks”, in *Handbook of Data Analysis*, Melissa Hardy, Alan Bryman , 1st ed. SAGE Publications, 2004, pp. 505-526.

[3] Albert-Laszlo Barabasi, *Linked: The New Science of Networks,* 1st ed. Basic Books, 2002.

[4] H.V. Alemann, *Der Forschungsprozess. Eine Einführung in die Praxis der empirischen Sozialforschung*, 1st ed. Stuttgart: Teubner , 2007.

[5] David Lazer, Alex Pentland, Lada Adamic, et. al., *Computational Social Science*, Science 323, 2009, pp. 721-724.

[6] A. Pentland, *Honest Signals: How They Shape Our World*, 1st ed. Cambridge,MA: The MIT Press, 2008

[7] Riitta Toivonen, Jukka-Pekka Onnela, Jari Saramäki, et. al., *A model for social networks*, Physica A 371, 2006, pp. 851-860.

[8] Julián Candia, Marta C. González, Pu Wanga, et. al.: *Uncovering individual and collective human dynamics from mobile phone records*, Journal of Physics A: Mathematical and Theoretical 41, 2008.

[9] R. Toivonen, J. M. Kumpula, J. Saramäki, *The role of edge weights in social networks: modelling structure and dynamics*, Proceedings of SPIE Vol. 6601, 66010B, 2007.

[10] Yong-Yeol Ahn, James P. Bagrow, Sune Lehmann, *Link communities reveal multiscale complexity in networks*, Nature 1038, 2010, pp. 1-5.

[11] Cesar A. Hidalgo, C. Rodriguez-Sickert, *The dynamics of mobile phone networks*, Physica A 387:12, 2008, pp. 3017-3024.

[12] Marta C. González, César A. Hidalgo, Albert-László Barabási, *Understanding individual human mobility patterns*, Nature 453, 2008, pp. 779-782.

[13] Gergely Palla, Albert-László Barabási, Tamás Vicsek, *Quantifiing social group evolution*, Nature 446:7136, 2007, pp. 664-667.

[14] Gregory R. Madey, Albert-László Barabási, Nitesh V. Chawla, et. al., *Enhanced Situational Awareness: Application of DDDAS Concepts to Emergency and Disaster Management*, Beijing, China: International Conference on Computational Science, 2007.

[15] Timothy Schoenharl, Ryan Bravo, Greg Madey, *WIPER: Leveraging the Cell Phone Network for Emergency Response*, International Journal of Intelligent Control and Systems, 11(4), 2006.

[16] Jukka-Pekka Onnela, Jari Saramäki, Jörkki Hyvönen, et.al., *Analysis of large-scale weighted network of one-to-one human communication*, New Journal of Physics 9, 2007, pp. 1-27.

[17] Lada Adamic, Eytan Adar: *How to search a social network*, Social Networks, 27(3), 2005, pp. 187-203.

[18] Christoph Loch, Joshua Tyler, Rajan Lukose, *Conversational Structure in Email and Face-to-face Communication*, Draft submitted to Organization Science. 2003

[19] Bernie Hogan, Danyel Fisher, *A scale for measuring Email Overload*, Microsoft Research Technical Report 2006-65, 2006

[20] Lada Adamic, Natalie Glance, *The political blogsphere and the 2004 U.S. Election: Divided They Blog*, LinkKDD ’05: Proceedings of the 3rd international workshop on Link discovery, 2005

[21] Scott Golder, Dennis Wilkinson, Bernado Huberman, *Rhythms of social interaction: messaging within a massive online network*, East Lansing MI: 3rd International Conference on Communities and Technologies CT2007, 2007

[22] Cameron Marlow(2009, March 9), *Maintained Relationship on Facebook* [Online].
Available: http://overstated.net/2009/03/09/maintained-relationships-on-facebook

[23] Mark Granovetter, *The strenght of weak ties*, American Journal of Sociology, 1973.

[24] Deni Khanafiah, Hokky Situngkir, *Social Balance Theory, Revisiting Heider's Balance Theory for many agents*, unpublished.

[25] Jure Leskovec, Daniel Huttenlocher, Jon Kleinberg, *Signed Networks in Social Media*, 28th ACM SIGCHI Conference on Human Factors in Computing Systems, 2010.

[26] Robert Kooij, Almerima Jamakovic, Frank van Kesteren, et.al., *The Dutch Soccer Team as a Social Network*, INSNA - Connections Volume 29 Issue 1, 2009, pp. 4-14.

[27] Guillaume Erétéo, Michel Buffa, Fabien Gandon, et.al., *A State of the Art on Social Network Analysis and its Application on the Semantic Web*, Karlsruhe, Germany: In Proc. SDoW2008 Social Data on the Web, Workshop held with the 7th International Semantic Web Conference, 2008

[28] Li Ding, Tim Finin, Anupam Joshi: *Analyzing Social Networks on the Semantic Web*, IEEE Intelligent Systems, IEEE Computer Society, 2004.

[29] Li Ding, Tim Finin, Anupam Joshi, et. al., *Swoogle: A Semantic Web Search And Metadata Engine*, Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management, 2004

[30] Peter Mika: *Flink: Semantic Web Technology for the Extraction and Analysis of Social Networks*, Journal of Web Semantics, Volume 3, Number 2, 2005

[31] Associated Press(2005, October 15), *Phone Tap: How's the Traffic?* [Online].
Available: http://www.wired.com/gadgets/wireless/news/2005/10/69227

[32] Rachel Metz(2005, November 8), *Saving the world with cell phones*, [Online].
Available: http://www.wired.com/gadgets/wireless/news/2005/08/68485

[33] Jon Kleinberg, *The wireless epidemic*, Nature (News and Views) 449, 2007, pp. 287-288

[34] Stanley Wasserman, Katherine Faust, *Social Network Analysis: Methods and Applications*,
1st ed. Cambridge University Press, 1994

[35] P.J. Carrington, J. Scott, S. Wasserman, *Models and Methods in Social Network Analysis*, Cambridge University Press, 2005

[36] Stallmann, Matthias (2009, June 2), *The 7/5 Bridges of Koenigsberg/Kaliningrad* [Online]
Available: http://www.csc.ncsu.edu/faculty/stallmann/SevenBridges/

[37] S. Boccaletti, V. Latora, Y. Moreno, et. al., *Complex networks: Structure and dynamics*, Physics Reports Vol. 424 Issues 4-5, 2006, pp. 175-308

[38] M. T. Goodrich and R. Tamassia, *Data Structures and Algorithms in Java*, 4th ed. Wiley, 2005

[39] Vladimir Batagelj, Andrej Mrvar, *Pajek – Program for Analysis and Visualization of Large Networks, Reference Manual*, Lublijana, 2010, Available: http://vlado.fmf.uni-lj.si/pub/networks/pajek/doc/pajekman.pdf

[40] James Moody, Douglas R. White, *Structural Cohesion and Embeddedness: A Hierarchical Concept of Social Groups*, American Sociological Review, Vol. 68, No. 1., 2003, pp. 103-127

[41] Michelle Girvan, M.E.J. Newman, *Community structure in social and biological networks*, Proceedings of the National Academy of Sciences of the United States of America, Vol. 99, No. 12., 2002, pp. 7821-7826

[42] M.E.J. Newman, *Modularity and community structure in networks*, Proceedings of the National Academy of Sciences USA 103, 2006, pp. 8577-8582

[43] Miller McPherson, Lynn Smith-Lovin, James M. Cook, Birds of a Feather: *Homophily in Social Networks*, Annual Review of Sociology Vol. 27, 2001, pp. 415-444

[44] M.E.J. Newman, *Mixing patterns in networks*, Physical Review E 67, 026126, 2003

[45] W3C (2010, August 05), *World Wide Web Consortium* [Online]
Available: http://www.w3c.org

[46] Tim Berners-Lee(2006, March 09), *Notation 3 specification* [Onlnine]
Available: http://www.w3.org/DesignIssues/Notation3.html

[47] W3C HTML 4.1 Specification, 1999
http://www.w3.org/TR/html4/

[48] RFC 1630, Unique Resource Identifiers in WWW

[49] RFC 2616, Hypertext Transfer Protocol – 1.1

[50] Tim Berners-Lee (2006-07-27) *Linked Data* [Online]
Available: http://www.w3.org/DesignIssues/LinkedData.html

[51] Tom Baker, Guus Schreiber, Ralph Swick, et. al.(2010, February 3) *Semantic Web Deployment Working Group* [Online]
Available: http://www.w3.org/2006/07/SWD/

[52] Tim Berners-Lee (2004 February 10) *RDF Primer* [Online]
Available: http://www.w3.org/TR/rdf-primer/

[53] Dan Brickley (2004, February 10) *W3C RDF Schema Specification* [Online]
Available: http://www.w3.org/TR/rdf-schema/

[54] Peter Patel-Schneider, Patrick Hayes, Ian Horrocks (2009 November 09) *W3C OWL Semantic and Abstract Syntax* [Online]
Available: http://www.w3.org/TR/owl-semantics/

[55] W3C (2010, September 6) *XML* [Online]
Available: http://www.w3.org/XML/

[56] OWL2 Quick Reference, 2009
Available: http://www.w3.org/TR/owl2-quick-reference/

[57] RDFa Primer, 2008
Available: http://www.w3.org/TR/xhtml-rdfa-primer

[58] Alistair Miles, Sean Bechhofer (2009, August 18) *SKOS Simple Knowledge Organization System* [Online]
Available: http://www.w3.org/TR/2009/REC-skos-reference-20090818/

[59] Eric Prud'hommeaux, Andy Seaborne (2008, January 16) *SPARQL Query Language for RDF* [Online]
Available: http://www.w3.org/TR/rdf-sparql-query/

[60] Richard Cyganiak (2010 September 22) *Linked Data Cloud* [Online]
Available: http://richard.cyganiak.de/2007/10/lod/lod-datasets_2010-09-22_1000px.png

[61] Pierre Bourdieu, *Ökonomisches Kapital - Kulturelles Kapital - Soziales Kapital* in: Reinhard Kreckel "Soziale Ungleichheiten", Göttingen, 1983, pp. 183-198.

[62] Eva Heller, Wie Farben wirken: Farbpsychologie. Farbsymbolik. Kreative Farbgestaltung, 5. ed. rororo; 2004

[63] Nina Baur, Hermann Korte, et al.: Handbuch Soziologie, VS Verlag, 2008

[64] Facebook Developers (2010, September 22) *Graph API reference* [Online]
Available: http://developers.facebook.com/docs/reference/api/

[65] Rasmus Hahn, Christian Bizer, Christopher Sahnwaldt, et. al., *Faceted Wikipedia Search*, Berlin, Germany: 13th International Conference on Business Information Systems (BIS 2010), 2010

[66] Fielding RT, *Architectural Styles and the Design of Network-based Software Architectures*.
PhD thesis, University of California, Irvine, 2000

[67] Auer, S., Bizer, C., Lehmann, J., et. al., *DBpedia: A Nucleus for a Web of Open Data*, Busan, Korea: The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, 2007.

[68] Sören Auer, Jens Lehmann, *What have Innsbruck and Leipzig in common? Extracting Semantics from Wiki Content*, Proceedings of European Semantic Web Conference (ESWC'07), 2007, pp. 503-517

[69] Lehmann, J.; Schüppel, J.; Auer, S., *Discovering Unknown Connections – the DBpedia Relationship Finder*, Leipzig, Germany: In Proceedings of 1st Conference on Social Semantic Web, CSSW2007, 2007.

[70] Christian Becker, Chrisitan Bizer, *DBpedia Mobile – A Location-Aware Semantic Web Client*, Karlsruhe, Germany: Semantic Web Challenge at ISWC 2008, 2008.

[71] Eick, S.T., *Aspects of network visualization*, IEEE Computer Graphics and Applications Volume 16 Issue: 2, 1996, pp. 69-72

Abbreviations

API	Application Programming Interface
CEO	Chief Executive Officer
CSCW	Computer Supported Cooperative Work
DOM	Document Object Model
EOL	End Of Line
FOAF	Friend of a Friend
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	HyperText Transfer Protocol
IMDB	Internet Movie DataBase
N3	Notation 3
OWL	Web-Ontology Language
RDF	Resource Description Framework
RDFS	RDF Schema
SKOS	Simple Knowledge Organization System
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WLAN	Wireless Local Area Network
XML	eXtensible Markup Language

Appendices

Appendix A

Appendix A contains an expert interview on social science and social network analysis.

A: Hallo, danke dass du dir Zeit genommen hast. Ich möchte für meine Masterarbeit ein paar Informationen über die Sozialwissenschaft sammeln und dir deshalb ein paar Fragen stellen.

B: Gerne.

A: Welche Kerngebiete sind deiner Meinung nach in der Sozialwissenschaft relevant?

B: Ich würde sagen auf jeden Fall die Soziologie! Die Soziologie beschäftigt sich mit der Gesellschaft und gesellschaftlichen Zusammenhängen, mit zwischenmenschlichen Interaktionen und Handlungen Ich denke die Soziologie ist wichtig weil sie sich mit der sozialen Welt beschäftigt, sie beschäftigt sich schließlich sogar auf einer Metaebene mit der Wissenschaft selbst, die Soziologie sieht sich als Sozialwissenschaft selbst als Bestandteil der sozialen Welt und der Gesellschaft. Aber viele andere Wissenschaften sind in der Sozialwissenschaft präsent, sie ist eine interdisziplinäre Wissenschaft. Auf jeden Fall Wissenschaften wie die Philosophie, die Kultur- und Sozialanthropologie, de Politikwissenschaft, die Kommunikationswissenschaft, aber auch Geographie und Geschichte würde ich dazu zählen.

A: Mit welchen Bereichen und Themen beschäftigen sich die Sozialwissenschaften?

B: Mit allen Bereichen des sozialen Lebens. Mit „großen“ Themen wie Globalisierung oder der Ökonomie und Wirtschaft, mit dem Staat an sich, oder der Transnationalität, der Auflösung des Nationalstaates. Aber eigentlich einfach mit der Gesellschaft und ihrer Politik, dem Rechtssystem, der Wirtschaft, Städtebau und Urbanität, den Menschen, Sub- und Jugendkulturen, der Religion. Aber auch mit Bereichen wie dem medizinischen Versorgungssystem, der Verteilung von Krankheit. Die Sozialwissenschaften beschäftigen sich mit allem Möglichen. Mit der Technik, der Wissenschaft selbst, dem Bildungssystem und der Bildung an sich, der Kunst und der Kultur, der, der Mode, dem Sport, dem Internet... alle Erscheinungen und Phänomene die in einer Gesellschaft beobachtbar sind, die in irgendeiner Form sozial erzeugt werden, können Gegenstand der Sozialwissenschaften sein.

A: Welche Methoden werden in der Sozialwissenschaft verwendet?

B: Ich kann hier in erster Linie nur für die Soziologie sprechen, aber im allgemeinen sind diese Methoden in der Sozialwissenschaft auch anwendbar und werden auch verwendet. Eine wichtige Unterscheidung ist die zwischen qualitativer und quantitativer Forschung. Die quantitative Sozialforschung beschäftigt sich sozusagen mit der gesamten Gesellschaft, sie versucht einige Merkmale an möglichst vielen Merkmalsträgern zu untersuchen. Also z.B. viele Menschen in Österreich über ihren Familienstand, Einkommen, Arbeitsverhältnis und demographische Daten befragen mit einem Fragebogen für ein Social Survey, um das etwas einfacher zu sagen. Quantitative Erhebungs- und Auswertungsverfahren sind stark strukturiert. Interviews und Befragungen, Beobachtungen diese Techniken kommen oft zum Einsatz. Auch Experimente, vor allem das Laborexperiment ist ein quantitatives Verfahren, weil ja alle Einflüsse konstant gehalten werden. In den Sozialwissenschaften kommen aber eher Feldexperimente vor weil sie nicht so künstlich sind. Dabei beobachtet oder untersucht man quasi im Feld, also im Leben selbst, im Alltag quasi. Die Auswertung der Daten sind dann meistens statistische Verfahren um später von der Stichprobe auf die Grundgesamtheit zu schließen. Die quantitative Forschung ist hypothesenprüfend, diese typischen wenn-dann-Hypothesen, wenn A dann B. Die qualitative Forschung ist hypothesengenerierend, man forscht eher in die Tiefe so zu sagen. Man versucht möglichst viel über einen oder einige wenige Fälle herauszufinden. Qualitative Techniken der Datenerhebung sind qualitative sehr offene Interviews, oft narrative Interviews und Befragungen, Beobachtungen, Artefaktanalyse. In der Auswertung dann qualitative Textanalyse, Diskursanalyse, hermeneutische Interpretation, es gibt da sehr viele Verfahren.

A: Kennst du die Technik der Netzwerkanalyse?

B: Ja, ich denke die Netzwerkanalyse kommt zu wenig zum Einsatz. Mit ihr kann man nachdem man Daten gesammelt hat recht rasch und übersichtlich quantitativ Gruppen darstellen und untersuchen - auch sehr große Gruppen. Man kann die Zusammenhänge und Interaktionen innerhalb der Gruppe, aber auch Zusammenhänge zwischen verschiedenen Gruppen darstellen. Ich beschäftige mich gerade mit Online-Kommunikationsplattformen und gerade da ist die Netzwerkanalyse sehr praktisch. Ich habe viele Fallstudien gefunden in denen es um die Kommunikationsnetzwerke und verschiedenen Freundeskreise der User ging, da wäre Netzwerkanalyse sehr hilfreich und wird soweit ich weiß auch verwendet. Es kommt einfach immer darauf an wie genaue Informationen und Erkenntnisse man erfahren und erhalten möchte, aber für eine grundsätzliche Darstellung und eine Untersuchung von großen Menschenmengen ist sie sicherlich ein gute Technik, man benötigt aber immer ausreichend Informationen über die Gruppen und deren Handlungen um sie anwenden zu können.

A: Ist an der Netzwerkanalyse deiner Meinung nach etwas Besonderes, dass sie von anderen Techniken und Methoden unterscheidet?

B: Jede Methode hat ihre speziellen Vorzüge... aber ich denke die Netzwerkanalyse ist die einzige in der es möglich ist eine große Gruppe in ihrer Gesamtheit darzustellen und zu erfassen. Also sowohl jedes Individuum, als auch jedes Individuum als Teil der Gruppe und auch die Verbindungen zwischen ihnen. Und das Ganze funktioniert ohne mit Wahrscheinlichkeitsrechnung und Irrtumswahrscheinlichkeit hochrechnen zu müssen. Würde ich die Nutzer einer Online-Kommunikationsplattform wie Facebook in Bezug auf ihre Kommunikationsmuster untersuchen wollen könnte ich mit der Netzwerkanalyse theoretisch einfach alle Netzwerke und Nutzer erfassen, statt nur eine Stichprobe zu untersuchen. Die Netzwerkanalyse ist in der Hinsicht sehr sinnvoll, das Problem ist allerdings an die benötigten Daten zu gelangen und Mitarbeiter aufzutreiben die Netzwerkanalyse an sich und auch die entsprechenden Computerprogramme dafür kennen und verstehen, weil diese Technik eben computerunterstützt ist, manuell wäre das unmöglich.

A: Gut, Dankeschön, das wären alle Fragen gewesen, danke dass du dir Zeit genommen hast.

B:Bitteschön.

A: Interviewer

B: Soziologiestudentin an der Universität Wien, kurz vor dem Abschluss

Interview vom 03.08. 2010

Appendix B

Appendix B shows the text of the supplement for the usability study:

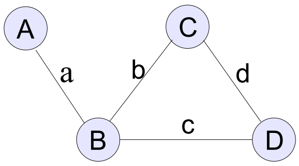
Danke, dass Sie sich dazu entschlossen haben, an der Studie, welche einen wichtigen Teil meiner Masterarbeit darstellt, teilzunehmen. Dieser Leitfaden erklärt Ihnen in 3 kompakten Teilen worum es in meiner Arbeit geht und zeigt Ihnen alle Facetten meines entwickelten Tools. Falls Sie sich nicht dafür interessieren, können Sie direkt zu den konkreten schritt-für-schritt Anwendungsszenarien auf Seite 4 springen.

Mein Tool bzw. meine Webseite bedient sich einer bestimmten Datenstruktur, Linked Data, und extrahiert daraus Soziale Netzwerke um sie zu analysieren. Die restliche Seite gibt Ihnen eine kurze Einführung in Soziale Netzwerke und Linked Data. Wenn Sie über diese zwei Punkte Bescheid wissen, dann überspringen Sie diese.

Teil 1 - Hintergrundwissen

Soziale Netzwerke

Die Soziale Netzwerkanalyse ist eine Methode, die auf der Graphentheorie basiert. In Abbildung 1 ist so ein graphentheoretisches Konstrukt dargestellt. A, B, C, D sind die Knoten (im engl. Vertices od. Nodes) und a,b,c,d sind die Kanten (im engl. Edges, Arcs od. Links).



Ungerichteter Graph

Knoten sind bei sozialen Netzwerken in der Regel Menschen. Es können aber auch Gruppen, soziale Klassen, wirtschaftliche Organisationen, Nationen, etc. sein. Kanten können gerichtet oder ungerichtet sein, ein und die selbe Bedeutung haben, oder aber auch verschiedene. Eine sehr bekannte Unterscheidung von Kanten in der Sozialwissenschaft ist die der Weak und Strong Ties von Mark Granovetter.

Eine andere gebräuchliche Unterscheidungsform ist die, um die Social Balance Theorie von Heider zu untermauern (Abbildung 2). Hier werden Kanten in Like und Dislike, oder + und -, unterteilt.

my friend's friend is my friend
my friend's enemy is my enemy
my enemy's friend is my enemy
my enemy's enemy is my friend

Heider's social balance Theorie

Linked Data

Linked Data ist eine bestimmte Art Daten für Webseiten oder andere Applikationen aufzubereiten. Diese Daten sind allgemein über das Internet direkt (also ohne den Umweg einer Webseiten-Präsentation) verfügbar. Zudem sind sie maschinen-lesbar und untereinander verlinkt.

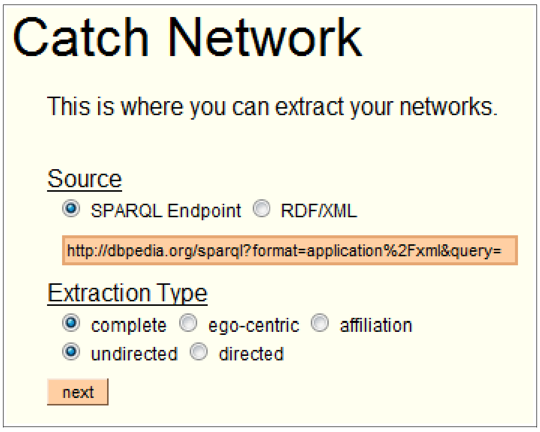
Zur Erklärung eignet sich am besten ein Beispiel aus der Praxis:

DBpedia ist ein Projekt, welches Daten aus der Wikipedia (das wiederum Daten nur über die Webseite anbietet, welche nur sehr eingeschränkt maschinenlesbar sind) extrahiert und aufbereitet. Konkret sind mittlerweile 3,4 Millionen Dinge aus der Wikipedia transferiert worden. Das sind 312.000 Personen, 413.000 Orte, 94.000 Musikalben, 49.000 Filme, 150.000 Videospiele, 140.000 Organisationen, 146.000 Spezien und 4.600 Krankheiten, und vieles mehr. Um so eine Resource zu betrachten klicken Sie bitte auf den folgenden Link um Informationen über Immanuel Kant einzusehen: http://dbpedia.org/page/Immanuel_Kant

Teil 2 – Catch Network

Klicken Sie auf der SocioCatcher Webseite auf den Menüpunkt „Catch Network“.

Hier kommen Sie im ersten Abschnitt auf eine Eingabemaske um einerseits die Quelle zu wählen, und andererseits um Einstellungen zu treffen, wie Ihr Netzwerk aussehen und extrahiert werden soll.



Catch Network

This is where you can extract your networks.

Source

● SPARQL Endpoint ○ RDF/XML

http://dbpedia.org/sparql?format=application%2Fxml&query=

Extraction Type

● complete ○ ego-centric ○ affiliation

● undirected ○ directed

SPARQL Endpoint

Geben Sie einen Endpoint einer bestehenden Linked Data Quelle ein. (DBpedia Endpoint ist per Default eingestellt)

RDF/XML

Laden Sie eine RDF/XML Datei auf den Server.(Nur für fortgeschrittene Nutzer)

Extraction Type

Bei Extraction Type geben Sie an, ob Sie ein komplettes Netzwerk (Bspw. ALLE Schriftsteller), ein Ego-Zentrisches Netzwerk (Bspw. Kant und seine umliegenden „Freunde“) oder ein Netzwerk über Gemeinsamkeiten(Bspw. ALLE Musiker, wobei die Musiker die bei den selben Plattenlables waren, „Freunde“ sind) extrahieren wollen. Zudem können Sie wählen, ob Ihr Netzwerk gerichtet oder ungerichtet sein soll.

Je nach Wahl erscheinen unterschiedliche Optionen um Ihre Knoten und Kanten zu wählen.

Nodes (Knoten)

Bei „complete“ und „affiliation“ bekommen Sie die Wahl einen bestimmten Knoten-Typ zu wählen. Beispielsweise geben Sie ein „rdf:type“ und „dbpedia-owl:Writer“ für alle Schriftsteller, oder „rdf:type“ und „dbpedia-owl:MusicalArtist“ für alle Musiker.

Bei „ego-centric“ geben Sie einen spezifischen Ego-Knoten ein, bspw. *http://dbpedia.org/resource/Immanuel_Kant* für ein Ego-Zentrisches Netzwerk von Kant.

Edges (Kanten)

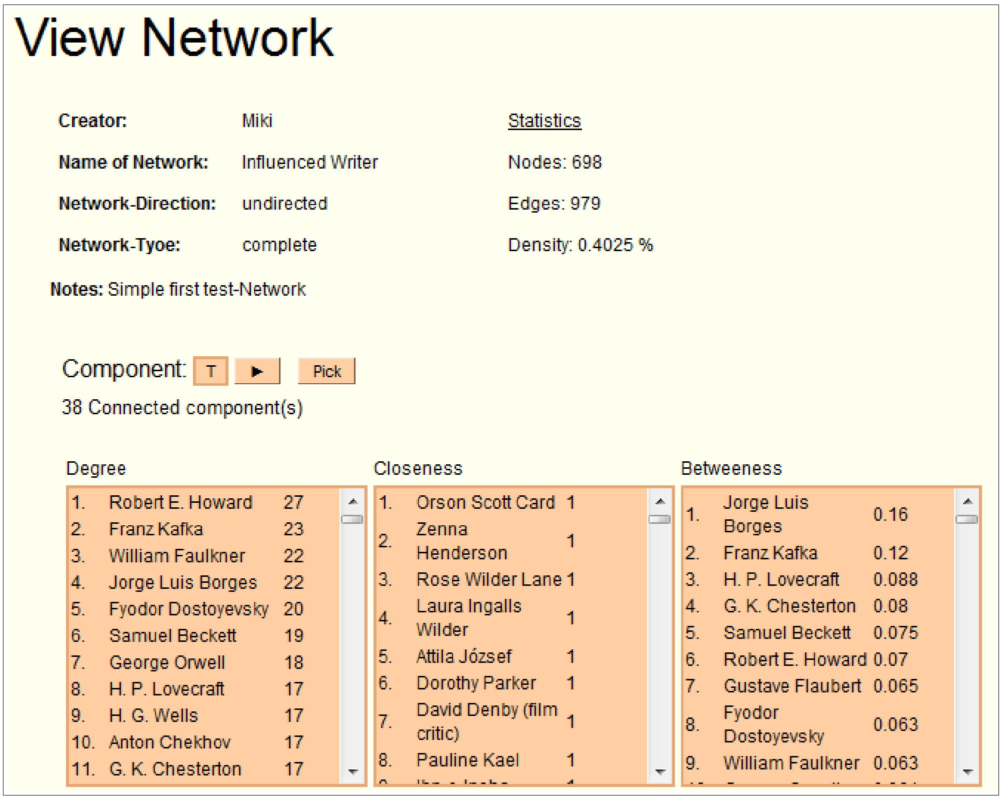
Bei „complete“ und „ego-centric“ wählen Sie die Kanten durch ein (oder merhere) bestimmte Prädikat(e), welches direkt auf ein anderes Objekt zeigt, das zu unserer Knotenmenge gehört. Bei einem Schriftsteller-Netzwerk kann man bspw. das Prädikat „dbpprop:influenced“ eingeben um dadurch Beziehungen (z.B.: welcher Schriftsteller hat wen beeinflusst) festzulegen.

Bei „affiliation“ werden die Kanten indirekt ermittelt. Man gibt eine Verbindung zu einer Organisation oder einem Event an, um daraus dann Gemeinsamkeiten und Freundschaften/ Bekanntschaften zu ermitteln. Bei einem Netzwerk von Musikern gibt man bspw. das Attribut „dbpedia-owl:recordLabel“ an um daraus ein Netzwerk zu schaffen in dem alle Musiker (die im selben Record Label waren) eine Verbindung zueinander aufweisen.

Um das Netzwerk ein wenig auszudünnen gibt es noch die Möglichkeit einer Grenze (Threshold), um bspw. zu sagen: „Alle Musiker die bei **2** gleichen Plattenlabels waren, sind „Freunde“ “.

Teil 3 – View Network

Wenn man ein Netzwerk extrahiert hat, kommt man auf die Seite „View Network“.



View Network

Creator: Miki
Name of Network: Influenced Writer
Network-Direction: undirected
Network-Tyoe: complete

Statistics
Nodes: 698
Edges: 979
Density: 0.4025 %

Notes: Simple first test-Network

Component:
38 Connected component(s)

Degree

1.	Robert E. Howard	27
2.	Franz Kafka	23
3.	William Faulkner	22
4.	Jorge Luis Borges	22
5.	Fyodor Dostoyevsky	20
6.	Samuel Beckett	19
7.	George Orwell	18
8.	H. P. Lovecraft	17
9.	H. G. Wells	17
10.	Anton Chekhov	17
11.	G. K. Chesterton	17

Closeness

1.	Orson Scott Card	1
2.	Zenna Henderson	1
3.	Rose Wilder Lane	1
4.	Laura Ingalls Wilder	1
5.	Attila József	1
6.	Dorothy Parker	1
7.	David Denby (film critic)	1
8.	Pauline Kael	1

Betweeneness

1.	Jorge Luis Borges	0.16
2.	Franz Kafka	0.12
3.	H. P. Lovecraft	0.088
4.	G. K. Chesterton	0.08
5.	Samuel Beckett	0.075
6.	Robert E. Howard	0.07
7.	Gustave Flaubert	0.065
8.	Fyodor Dostoyevsky	0.063
9.	William Faulkner	0.063

Diese Seite zeigt an wieviele Knoten und Kanten das Netzwerk hat, welche Netzwerkdichte es besitzt und auch wieviele Komponenten (also zusammenhängende Elemente) so ein Netzwerk besitzt und welche Komponente gerade angezeigt wird (T für Total network).

Ungerichtete Netzwerke haben Anzeigen zu den drei verschiedenen Werten, Degree(Grad jedes einzelnen Knoten), Closeness(wie nah ist der Knoten zu allen anderen Knoten) und Betweenness(wie wichtig ist der Knoten, wenn andere Knoten sich gegenseitig im Netzwerk erreichen wollen). So kann man sehen, welcher Knoten sehr zentral ist, oder welcher eine strategisch wichtige Position einnimmt. Closeness und Betweenness haben erst eine Aussagekraft, wenn man zusammenhängende Komponenten ansieht.

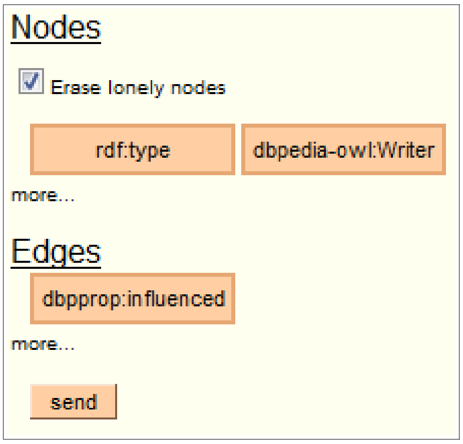
Gerichtete Netzwerke haben die Werte Indegree, Outdegree und Page Rank. Indegree und Outdegree zeigt an wieviele Kanten zu einem Knoten hin bzw. weggehen und Pagerank ist ein komplizierterer Algorithmus um wichtige Knoten zu erkennen. Der Grundgedanke von Pagerank ist der, dass Knoten die viele „stimmen“ erhalten, die also wichtig für das Netzwerk sind, das deren Stimme wiederum mehr zählt wenn diese für jemanden stimmen.

Zusätzlich gibt es die Möglichkeit das Netzwerk in dem für Netzwerke üblichen Pajek-Format (ein frei verfügbares Programm zur Netzwerkanalyse) auf die Festplatte runter zu laden um zusätzliche Analysen durchzuführen zu können und/oder es zu visualisieren.

Konkrete Anwendungszenarien

Influenced Writer

1. Klicken Sie auf „Catch Network“.
2. Lassen Sie alles unverändert und klicken Sie auf den Button next.
3. Machen Sie ein Häkchen bei Erase lonely nodes, geben bei dem Feld „type...“, „dbpedia-owl:Writer“ ein und bei dem Feld „predicate...“, „dbpprop:influenced“ und klicken auf den Button send.



Nodes

☑ Erase lonely nodes

rdf:type | dbpedia-owl:Writer

more...

Edges

dbpprop:influenced

more...

send

Nach einer kurzen Verarbeitungszeit sehen Sie das fertige Netzwerk und können sich mit ▶ durch die einzelnen Komponenten klicken.

Musicians at the same Label

1. Klicken Sie auf „Catch Network“.
2. Klicke bei Extraction Type auf Affiliation, lasse den Rest so wie er ist und klicke auf next.
3. Mach ein Häkchen bei Erase lonely nodes, ändere das Feld „type...“ zu „dbpedia-owl:MusicalArtist“ und das Feld „relation...“ zu „dbpedia-owl:recordLabel“. Den Grenzwert kann man bei zwei belassen, oder auch auf 3 setzen, wenn man möchte. Klicke auf send.

Wenn Sie eigene Netzwerke erstellen wollen, dann empfehle ich Ihnen sich genauer mit DBpedia auseinander zu setzen. Mögliche weitere Szenarien die mit dem Tool zu extrahieren wären:
„Alle Fußballer die bei den 3 gleichen Vereinen gespielt haben, sind miteinander befreundet.“
„Alle Künstler haben sich gegenseitig beeinflusst“
„Alle Philosophen mit dem gleichen Geburts- oder Todesort haben eine Verbindung zu einander“
… …

Bei anderen „SPARQL Endpoints“ liesen sich auch sogenannte Co-Authorship Networks extrahieren. Das heißt, dass laut Netzwerk alle (wissenschaftlichen) Autoren, die gemeinsam publiziert haben, miteinander befreundet sind. Den Möglichkeiten sind also keine Grenzen gesetzt.

Wenn Sie sich weitgehender mit der Thematik befassen möchten, dann bitte schreiben Sie mir doch auf miki.zehetner@gmail.com. Ich würde mich freuen.

Danke, dass Sie sich Zeit genommen haben,
Miki Zehetner

Appendix C

Appendix C describes the multimedia supplement (CD) of this thesis:

Folder Networks

- Network-*.net	the analyzed Pajek-Net Networks
- Statistics.ods	OpenOffice Calc file for additional Statistics

Folder Software

- Project Data	additional design supplements
- Server	complete server data

Folder Thesis and Documents

- Usability Study	this folder contains all filled-in questionnaires as PDF files (german)
- Supplement.pdf	supplement used for the usability study, with use case scenarios (german)
- Installation.pdf	a complete installation guide (german)
- Interview.pdf	interview with sociology student about social science and social network analysis
- Expose.pdf	expose at the start of the thesis
- masterthesis.pdf	master thesis

Curriculum Vitae

Miki Alvin Zehetner

Student at
University of Vienna, Faculty of Computer Science
Vienna University of Technology, Faculty of Informatics

Personal

Born:	October 21, 1985 in Vienna, Austria
Citizenship:	Austria

Education

2010 -	Master in Didactics of Informatics
2010 -	Master in Media Informatics
2007 - 2010	Bakkalaureat in Computer-Management
2000 - 2006	HTBLVA Spengergasse for Electronic Data Processing and Organization

Professional experience

Tutor at University of Vienna, Department of Knowledge and Business Engineering

Marginally employed at Astro Experts, responsible for online representation and online shop

Trainee at VA Tech SAT, department of Software Engineering / department of Training and Human Resources