Evaluation of the Impact of Engineering Education Research Grants Using Software Tools: A Foundation

An Interactive Qualifying Project proposal to be submitted to the faculty of Worcester Polytechnic Institute in partial fulfillment of the requirements for the Degree of Bachelor of Science

December 16, 2010

Submitted by:

Ningxin Fu
Craig Janeczek
John Kimball
Kaitlyn Spetka

Submitted to:

Project Advisors:
Prof. Hanlan
Prof. Mathews

Project Liaisons:
Ms. Sue Kemnitzer
Dr. Alan Cheville

Abstract

The goal of our project was to provide the NSF with a software suite which evaluates the impact of engineering education research grants. We assisted the NSF by identifying interactions that influence the impact of grants and any measureable data within these interactions. We presented three deliverables and examined software tools to collect, organize, analyze, and visualize the quantifiable data within the interactions. Our endeavors serve as a framework for future investigation into grant impact evaluation.

Executive Summary

Since both developing and developed countries, such as the United States, depend on having well-educated engineers, engineering education is a major concern for most industrially advanced countries. There are organizations in existence, such as the Accreditation Board for Engineering and Technology (ABET) in the United States, to ensure that engineers in the United States receive quality, standardized education. Since the level of performance of a newly graduated engineer is based more heavily upon the engineer's education than "real world" experience in their chosen fields, the quality of engineering education is of utmost importance both to businesses and industry and to the success of the global economy.

The National Science Foundation (NSF) has invested a considerable amount of money into engineering education through its Engineering Education and Centers Division (EEC). However, the budget of the NSF, and specifically the EEC, is small in comparison to the total amount of money invested in engineering education in America (personal communication, Alan Cheville). Therefore, the EEC needs to be able to optimize the impact its money has on engineering education research. To do this, the EEC needs to be able to determine which researchers and institutions will be able to give it the biggest positive impact for the money.

The EEC requested that we create a foundation for a system that will allow the EEC to evaluate the impact of its portfolio on the field of engineering education research. We decided to complete this task by creating plans for effective procedures to analyze Principal Investigator (PI) and co-Principal Investigator (co-PI) networks, authorship networks, and the evolution of curriculum as described in college course catalogs over time. Each plan

contained a suite of software tools for the compilation, organization, analysis, and visualization that could efficiently be applied to each set of data. Although the plans we developed were specific to our deliverables, the software tools we analyzed could easily be applied to many different types of data, as long as the data are in a format compatible with the software tool.

Our investigation led us to conclude that there were distinct analysis plans for each deliverable. The analysis of PI and co-PI networks primarily used a Database Management System (DBMS) and social network analysis tools. This was similar to the analysis of the authorship network, which utilized the same tools but a different database. The analysis of course descriptions involved the use of a web crawler and text analysis tools. These specific uses for these types of software are a small sample of the potential uses for any of these software tools, and we strongly encourage further exploration into additional uses.

Acknowledgements

We would like to thank four people who greatly assisted us with the development of this project: Professor James Hanlan, Professor Lauren Mathews, Dr. Alan Cheville, and Ms. Sue Kemnitzer.

Professor Hanlan and Professor Mathews have been providing us invaluable assistance in our project development. They provided constant suggestions and critiques that were helpful for all of the sections of the project report. They also constantly assisted us in developing new ideas throughout our project. We are grateful for their assistance and efforts.

Dr. Cheville provided guidance throughout the development of our project at the National Science Foundation (NSF) in Arlington, VA, and always helped us with any problems we approached him with. He did this even though he was busy with meetings, panels, and other responsibilities. He also offered us a number of opportunities for attending conferences, lectures, and webinars that were interesting and relevant to our project. His support and guidance are much appreciated.

Ms. Kemnitzer made us feel warmly welcomed at the NSF. She took the time to hear about our project multiple times and arrange several opportunities for us to meet with WPI alumni at the NSF. We appreciate her help immensely.

We would also like to thank everyone else who has shared valuable professional opinions and knowledge with us. There are far too many other people to name individually, but we appreciate the help of every one of them.

Table of Contents

Table of Tables

Table of Figures

1.0 Introduction

Because both developing and developed countries, such as the United States, depend on having well-educated engineers, engineering education is a major concern for most industrially advanced countries. There are organizations in existence, such as the Accreditation Board for Engineering and Technology (ABET) in the United States, to ensure that engineers in the United States receive quality, standardized education. Since the level of performance of a newly graduated engineer is based more heavily upon the engineer's education than "real world" experience in their chosen fields, the quality of engineering education is of utmost importance both to businesses and industry and to the success of the global economy.

Realization of the importance of engineering education has led to research specifically geared toward evaluating engineering education. Rapid advances in science and engineering mean that engineering education is continuously evolving and therefore must be constantly monitored and evaluated (Kim, 2010). The National Science Foundation (NSF), based in Arlington, Virginia, has directed the flow of federal money into many aspects of science and engineering research. In particular, the Engineering Education and Centers Division (EEC) of the NSF has granted funds for many research projects in engineering education. However, the budget of the NSF, and specifically the EEC, is small in comparison to the total amount of money invested in engineering education in America (personal communication, Alan Cheville). Therefore, the NSF needs to be able to optimize the impact its money has on education. This means that the EEC has particular interest in optimizing its impact on the field of engineering education research. To do this, the EEC needs to be able to determine which researchers and institutions will

be able to give it the biggest positive impact for the money. The NSF currently requires that each awarded grant employ assessment mechanisms in order to collect specific data for the evaluation of impact or effectiveness of the research. The NSF would like an internal system of software tools that would allow it to analyze the raw data of its current evaluation metrics, as well as other evaluation metrics which take into account more abstract connections, such as the institutionalization of new ideas and practices that were produced as a result of an NSF grant, in order to better judge the effectiveness of the NSF's grants. Although their current assessment mechanisms produce reliable data, the EEC's current process of analyzing the data is expensive, labor intensive, and restricted by the current evaluation metrics.

The EEC funded the Science and Technology Policy Institute (STPI), a group which exists to support the executive branch by providing analytical data to policy makers, to hold two workshops in back-to-back years (Science and Technology Policy Institute, 2009). In 2008 the topic was evaluation of engineering education research (Balakrishnan, Lal, & Flattau, 2008) and in 2009 the topic was the development of strategies to evaluate research in Engineering Education (Lal, 2009). These workshops outlined how the field of engineering education will ideally progress in the coming years and proposed methods for quantifying the effectiveness of such educational research to make sure that engineering education does not go in the opposite direction. The most prevalent methods of evaluating the impact of the work of a researcher are based upon the ratio of the number of times that an author is cited to the number of papers that he or she has written. The workshops also outlined methods that assist with the collection of raw data from educational research programs.

Although research has been conducted on evaluating the effectiveness of a researcher through his or her publications, not many other approaches of evaluating effectiveness that apply to engineering education have been considered. There are many ways in which knowledge or technology resulting from an EEC research grant can impact the field of engineering education that cannot be calculated by examining the number of publications produced by the researchers. One general example of this impact is when a professor teaches using a methodology which was developed through the research funded by an EEC grant. There could be a paper that describes the teaching method developed as a result of the grant, but there would be no true way to gauge how effectively that method helps students learn better. Another metric not currently taken into account is the spread and evolution of the knowledge and technology stemming from an EEC grant after the submission of the final assessment to the NSF. For example, one EEC grant resulted in the development of an electronic board which aided the teaching of basic electrical engineering theories, along with a completely hands-on method of teaching with no traditional lectures. In the years after the grant ended, the website that detailed the electronic board and the associated teaching method had visits from faculty at many other colleges which eventually purchased some of the boards and adopted the teaching strategy. The popularity of the board eventually led to an electrical component company developing a similar piece of technology and selling it. This board made a significant impact on electrical engineering education but this impact would not be apparent by looking at papers published by the researcher who developed the board (personal communication, Don Millard).

The purpose of this project was to create a foundation for identifying which software tools work best for evaluating the impact of grants. To accomplish this, we identified some metrics that would allow the EEC division of the NSF to evaluate the impact of a grant

more completely, and recommended a plan to implement tools for those metrics. First we compiled a subset of the possible pathways of knowledge in order to determine a more complete way to evaluate impact of EEC grants. We then identified available software tools for each pathway that could be used to evaluate the impact of a grant. This gave the NSF a selection of software tools which could be immediately implemented in order to provide a multi-dimensional evaluation of a grant. We also recommended what we believe to be the best strategy to more comprehensively evaluate the impact resulting from a grant. With our successful analysis and recommendations, the NSF has a foundation for further investigation into grant impact analysis. Therefore, the NSF should be able to achieve its goal of using software tools to evaluate a more complete measure of impact of their grants.

2.0 Background

If one were to walk into an engineering classroom today, the professor might stand in front of a computer, showing sophisticated, computer generated graphs with a video projector. The students may copy down the notes onto a laptop or an electronic writing pad. Other classrooms are more simplistic, the professor only has chalk and a blackboard to lecture with; the students likely have to solve problems with only a pencil, paper, and a calculator. However, rapid advancements in engineering education as a result of engineering education grants help to develop the technologically driven society we live in today.

In recent years, there have been signs of change in engineering education. As there are many students who wish to pursue careers in engineering each year, the engineering community has recognized the importance of improving methods in engineering education to prepare future graduates for the rapidly evolving technical environment in the 21st century. This chapter will explain why it is important to evaluate the impact of National Science Foundation (NSF) grants and the various ways that evaluation is carried out.

2.1 The NSF and Support of Engineering Education

The NSF offers support for research in science and engineering education. Its mission is "to promote the progress of science; to advance the national health, prosperity, and welfare; to secure the national defense" (NSF-At a Glance, 2010). The NSF fulfills its mission chiefly by issuing limited-term grants. The annual budget of the NSF is nearly $6.9 billion in grants. As many as 10,000 grants can be given out per year (NSF-At a Glance, 2010). The Division of Engineering Education and Centers (EEC) plays an

important role in supporting engineering education by carrying out its objective: "to graduate engineering leaders with a global outlook and the ability to adapt to the rapidly evolving technical environment in industry, academe, and society." Like other divisions of the NSF, the EEC supports research in engineering education by issuing grant awards to institutions in the United States.

Even though the annual budget of the NSF is a sizable amount of money by itself, the budget is small compared to the total amount of money invested in education in the United States. In fact, its budget is one-tenth of a percent relative to the rest of engineering education money (personal communication, Alan Cheville). Therefore, the NSF, specifically the EEC, is trying to optimize the impact its grant money has on engineering education. In other words, the EEC constantly seeks effective uses of its grant money.

2.2 Evaluation of Programs

In order to achieve an efficient use of grant money, each program of research funded by the NSF is evaluated based on certain principles. Evaluation is a useful tool to measure the impact and the success of a project, as well as to measure the effectiveness of the grant money used. When NSF money is used well, it makes a positive impact. A positive impact is some sort of change that meets the NSF's goals of funding the given project. Furthermore, evaluation of NSF funded grants is necessary to provide a baseline to allow the NSF to make future decisions based on past projects.

2.2.1 Why is Evaluation Important to the NSF?

Primarily, evaluation provides information to help improve future projects. Information on whether certain goals are being met in the current stage of a project is

essential to the improvement process of a project. Evaluation methods have been developed to relate evaluation to program implementation (Frechtling, 2002). These methods have been developed to incorporate evaluation at the start of the program implementation. Evaluation should provide continuous feedback to any modification of the project at any time after the project has been implemented.

Secondly, evaluation provides a form of communicating results to those who fund the project (Frechtling, 2002). As a federal agency, NSF is required to report annually on the accomplishments of their funded efforts in accordance with the Government Performance and Results Act (GPRA) to the Senate Committee on Governmental Affairs. People responsible for NSF funded projects are asked to report back to the NSF on what they have actually done to impact various fields of research. These areas are supporting a workforce of scientists, engineers, and citizens. All of these people also work toward discovering innovations in many fields and share research well (National Science Foundation, 2001). Because of this, program directors need to have the GPRA and these outcomes in mind while developing plans for project evaluation.

2.2.2 Types of Evaluation

There are two types of evaluation that should be considered for NSF's programs: formative evaluation and summative evaluation (Frechtling, 2002). Formative evaluation is used in the early stages of a project to evaluate initial and ongoing project activities. In contrast, summative evaluation is used in the later stages to assess the overall impact and quality of a completed project (See Figure 2.1).

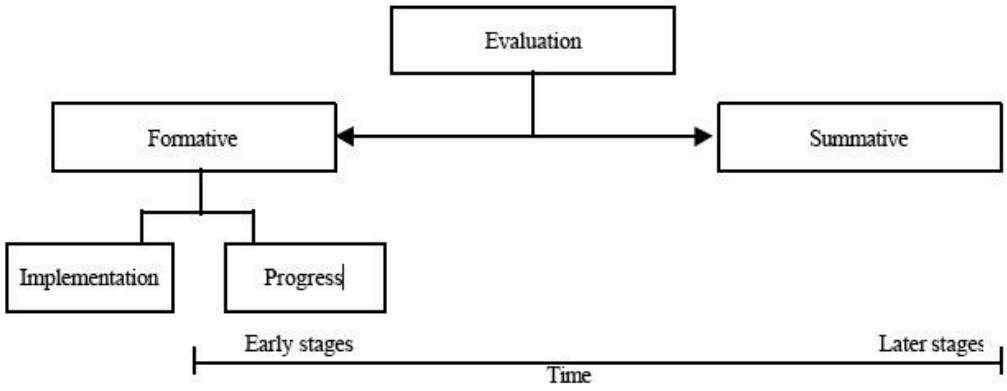

Figure 2.1: Types of Evaluation (Frechtling, 2002, p. 8). A timeline suggests when to use formative evaluation and summative evaluation within a project. As shown, formative evaluation is used during the early stages of a project, and monitors the project's progression from implementation to conclusion. Summative evaluation is used after the conclusion of the project, and determines the relative quality of the project.

The purpose of summative evaluation is to assess the success of the project in reaching its intended goals (Frechtling, 2002). Summative evaluation takes place after the project has been completed in the later stages of the diagram shown in Figure 2.1. Summative evaluation collects information about outcomes and related processes, strategies, and activities that have led to them. From a different standpoint, summative evaluation can also be addressed as a case study in which the entire program or NSF grant product may be considered as a whole (Stufflebeam & Shinkfield, 2007). The following basic questions may be addressed for a summative evaluation: does the funded project meet its goals of impact? If so, what are the most impactful components of the project? When the results of the program are achieved, when the program is finished, do the program's benefits outweigh its costs? Does this program seem to have fruitful future? Can this program be used in other places and situations? A student's final grade in a class is an example of summative evaluation because the grade is an evaluation of a student's cumulative performance in the class that is assigned once the class is complete.

The NSF should be concerned with summative evaluation because they need to know which grants have had a positive impact and which have not. Summative evaluation is mostly concerned with the impact of the end result, and the NSF uses summative evaluation after the program has been implemented to see if the grant has been impactful. If the given program has been impactful, the NSF will be more prone to give a grant to the owner of the funded program in the future.

Based upon the aforementioned concept of impact, the program director also has to have a few more specific questions. The answers to these questions may directly or indirectly affect decision-making by program officers. The main choices are the following: continue funding, increase funding, modify the given program and attempt implementation again, or discontinue or try the program somewhere else (Frechtling, 2002).

2.2.3 Evaluation Process

How is an evaluation carried out exactly? The answer to this question can be summarized into a couple main phases of evaluation. First, the person doing the evaluation develops a conceptual model of each program, and what can be evaluated from the program. Next, the evaluator outlines what the reasonable outcomes of each program are and formulates questions to gather that given data. Finally, the evaluator collects and analyzes the data, and relays the information of the outcomes to those who are interested (Frechtling, 2002).

A conceptual model is essential to every proposed evaluation. This model can provide common understanding amongst all Principal Investigator (PI)'s about the project's structure, connections, and expected outcomes by making these easy to visualize. Figure 2.2 presents a conceptual model called a "logic model" which describes individual elements

of the project, the connections among them, and the expected outcomes. A typical logic model has four categories of project elements: project inputs, activities, short-term outcomes and long-term outcomes (Frechtling, 2002). Describing a project in terms of the logic model is the first step of evaluation.

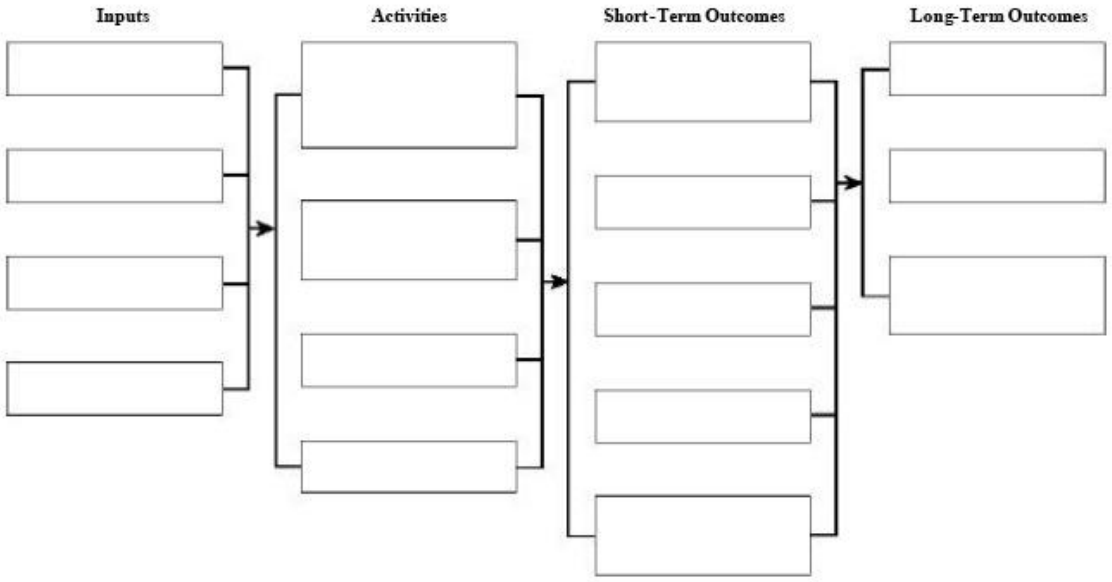

Figure 2.2: Logic Model (Frechtling, 2002, p. 16). A typical logic model has four categories of project elements: project inputs, activities, short-term outcomes and long-term outcomes (Frechtling, 2002). Describing a project in terms of the logic model is the first step of evaluation.

Two ideas must be kept in mind when designing this logic model. The first is determining who is funding the program and the second is which students are affected. These ideas are developed for the people who support the program because those people want to see the program's outcomes meet its objectives. Also, defining and prioritizing reasonable outcomes for each program are essential portions of the logic model. These are all key items in conceptualizing the logic model.

Developing the logic model also involves determining who will be studied and when they will be studied (Frechtling, 2002). There are two general approaches: quantitative (numbers) and qualitative (words), which are often used alternatively. In order to have a complete understanding of how well a project is meeting its expected goals, a "mix-method" design is often used to combine quantitative and qualitative techniques.

Developing such a design often requires sampling, use of comparison groups, timing, sequencing, and frequency of data collection. To study each, we must use a combination of qualitative and quantitative analysis which will dictate how well a project has met its goals in terms of impact. When addressing how to study the subjects, one must consider comparing groups of subjects and how frequently data is collected. For instance, one might notice a more prominent impact when two groups are analyzed frequently over a period in time compared to one analysis in time (Stufflebeam & Shinkfield, 2007).

The final steps vary greatly from project to project and have no set design to them. By using the materials produced in the first steps as guidelines, the most logical data collection process will become apparent. By using the evaluation method defined earlier, analyzing the collected data will provide the impact for the output of the project. The final step is summarizing what was done and the information produced by the project.

2.2.4 Current Evaluation Tools

The Engineering Education and Centers (EEC) Division of the NSF funded the Science and Technology Policy Institute to conduct a workshop on evaluation of research in 2008. The workshop attendees came up with an evaluation plan for research in engineering education in 2009. The evaluation tools that were developed could be categorized into one of two groups. The tools either determined the effectiveness of the individual performing

the research or the judged the quality of the application of the research. These tools are currently in use within the NSF.

The parameters evaluating an individual researcher's performance are known as the h-index, g-index, hc-index, hi-index, hi norm, Age Weighted Citation Rate (AWCR) and AWCR-index (Balakrishnan, Lal, & Flattau, 2008). Analyzing these individual researcher's performance provides a better viewpoint of the researcher's ability to produce quality research. All of these tools provide a tangible representation of how widely recognized and valued an individual's research is by looking at how the number of times any given work was cited. Indices are an example of tools currently used by the NSF for evaluation, but the reliance on indices for evaluation purposes results in a narrow view of the impact of a grant.

The second group of tools evaluates the quality of the application of the research. These tools consist of Innovation Indicators, Science Impact, Survey Data Collection, Case Study Analysis, Curriculum Vitae (CV) Analysis, and Data Mining. These tools provide an easier way to view the impact of research (Lal, 2009).

These are all currently used tools within the NSF. They are all good tools for evaluating impact, but they only take a small subset of data into account. All of these tools take a narrow view of the evaluation of the impact of a grant. Using these tools along with others to look at different data for the evaluation of the impact of a grant will yield a broader view of impact.

2.3 Data Collection

The term "data" refers to qualitative or quantitative attributes of a variable or set of variables. As the names imply, qualitative data are recorded or measured to describe the quality of something in size, appearance, or value, while quantitative data relate to a measure of the quantity of something. For the purpose of evaluating NSF sponsored projects, both qualitative and quantitative data collection methods are used to balance the trade-off between breadth and depth. For example, one particular student's transcript would give a broad quantitative representation of that student's academic performance. A qualitative survey asking the professor how the student has performed during his or her academic years would provide in-depth details about the academic performance of this student. Data collected through quantitative methods are often more objective and accurate because they can be collected using standardized methods. These methods can often be replicated and are generally not as sophisticated as qualitative data collection techniques can be (Frechtling, 2002). Qualitative data are most suitable for formative evaluations which are often done during the progress of a project because this type of data provides detailed information for how improvements can be made. Quantitative data are more suitable for summative evaluation because of the ability to capture the broad scope of the project; this provides a comprehensive overview after the project is finished. For the purposes of our project, we are seeking a way to use summative evaluation to judge not only the value of a project quantitatively, but also the impact of it. Therefore, a mixed method, the combination of quantitative and qualitative techniques, is the best approach to evaluate the impact of a funded project. The proceeding sections are going to introduce the methods of collecting and managing data that are applicable to our project.

2.3.1 Databases

A database is a collection of data (Silberschatz, Korth, & Sudarshan, 2001). Databases are widely used and can be found in fields ranging from banking to engineering. One example of a database is a human resources department which has information about employees, salaries, payroll taxes and benefits stored within different levels and partitions of a database. Databases are essential to almost all enterprises.

A database-management system (DBMS) provides an interface between a database and the user. A DBMS creates this interface by providing a convenient and efficient way to manipulate the data stored within the database. In order to provide the user a safe way to store and retrieve the data, the DBMS needs to define consistent structures for the storage of information within the database. Since the DBMS is able to do this effectively, it efficiently manages a large amount of information that is stored within a database.

As the World Wide Web has rapidly evolved since the 1990s, databases have been largely converted into web interfaces. A common example is online banking. When you access a bank website and retrieve your bank balance and transaction information, the information is retrieved from the bank's database system. Furthermore, when you visit a website, information about you may be retrieved from online databases at the same time, so that advertisements that are considered relevant to you are displayed. Data about your accesses to websites may also be stored in a database (Silberschatz, Korth, & Sudarshan, 2001).

2.3.2 Web Crawlers

One way of rapidly acquiring data from the World Wide Web is through the use of a web crawler. A web crawler is a program that takes advantage of the branch structure of

the World Wide Web by moving from page to page while collecting information (Pant, Srinivasan, & Menczer, 2004). A web crawler works by searching web pages for specific information and downloading it to an offline database. The flexibility of a web crawler allows the person designing it to tell the program exactly what to search for within each page.

In order to start the crawling process, the programmer must provide the crawler with a list of URLs to visit, which is also called a seed page. The crawler will then look through the seed page for any other pages which are linked to it and eventually search all of the pages found. As the web crawler searches the pages for more links to go to, it is able to download data such as images, videos, or even a copy of the page itself.

Due to the dynamic properties of the World Wide Web, there is always a need to update the offline copies of information. Therefore, the owner of the web crawler wants the web crawler to update the offline information as often as possible. Although this would prove to be the best strategy for the owner of the web crawler, this could slow down or even crash the web site that the crawler is searching. In order to protect the website owners, there are a number of protocols put in place by the owner of the web crawler: the selection protocol, the politeness protocol, and the revisit protocol. The selection protocol details which web pages the web crawler should search and the order of searching. The politeness protocol contains guidelines which attempt to minimize the harmful effects of a web crawler on the web, such as crashing a web page. The revisit protocol explains how to obtain the most up to date information from the web.

2.4 Data Analysis

Analysis of data is a process of organizing, transforming, and modeling raw data into usable information (Statistics Canada, 1998). Currently, data analysis is often closely linked to data visualization. Data visualization is an excellent way of presenting data because it can deliver information in an effective way without losing in-depth insights. Information visualization has been widely used and is generally applied to the visual representation of large-scale collections of non-numerical information (Friendly & Denis, 2001) . The major goal of data visualization is "to visualize data, communicating information clearly and effectively" (Friedman, 2008). For the purposes of this project, the type of data analysis that we focused on was social network analysis.

2.4.1 Social Network Analysis

Social Network Analysis is a powerful data analysis tool, which involves linking individuals through their interdependencies on other individuals. In order to examine these interactions, in 1954, J.A. Barnes began using the term "Social Network Analysis" to denote the links among people of different groups and how those people fall into bounded groups and social categories (Freeman, The Development of Social Network Analysis: A Study in the Sociology of Science, 2004). The public's interest in social network analysis has increased tremendously over the past generation (Knoke & Yang, 2008). The idea of a "small world" was especially popularized after the release of John Guare's film Six Degrees of Separation in 1993. This concept refers to the idea that everyone on Earth can be linked to anyone else within approximately six steps. Over the past three decades, social network analysis became an institutional concept and has been widely used by analysts in various fields such as sociology, anthropology, business management, biology, and so on.

Data that are used for social network analysis are quite different from conventional social science data. Conventional data are made of a rectangular array of measurements. The rows of the array are the subjects, and the columns of the array are variables. Each cell of the array provides information on the score of the actor (row) on the attribute (column) (Hanneman & Riddle, 2005). Table 2.1 shows an example of conventional rectangular data array.

Table 2.1: Example of Rectangular Data Array

Personal Information			
Name	Age	Nationality	Sex
Fleur	22	France	Female
Corinna	11	Germany	Female
Arne	30	United States	Male

Network data consist of a square array of measurements (Hanneman & Riddle, 2005). The rows of the array are the subjects, and the columns of the array are the same set of subjects. Each cell of the array describes the relationship between the pair of actors. An example is shown in Table 2.2, in this case, the relationship between the actors is a binary measure where a "0" indicates "No", and a "1" indicates "Yes".

Table 2.2: Example of Square Data Array

Who would you like to conduct research with?			
Chooser / Choice	Fleur	Corinna	Arne
Fleur	------	0	1
Corinna	1	------	0
Arne	1	0	------

The major difference between conventional data and network data is that conventional data focus on actors and attributes, while network data focus on actors and the relationships or connections among them. Analysis of network data could therefore be helpful in studying how a selected group of individuals is connected with each other. One example of network data analysis that is closely related to our project could be the co-authorship map. Examining how the PIs who receive NSF grants are connected with the rest of the scientific/engineering community gives an overall idea of how much impact the grants have had on this particular community. Figure 2.5 shows an example of a co-authorship map of social network scholars.

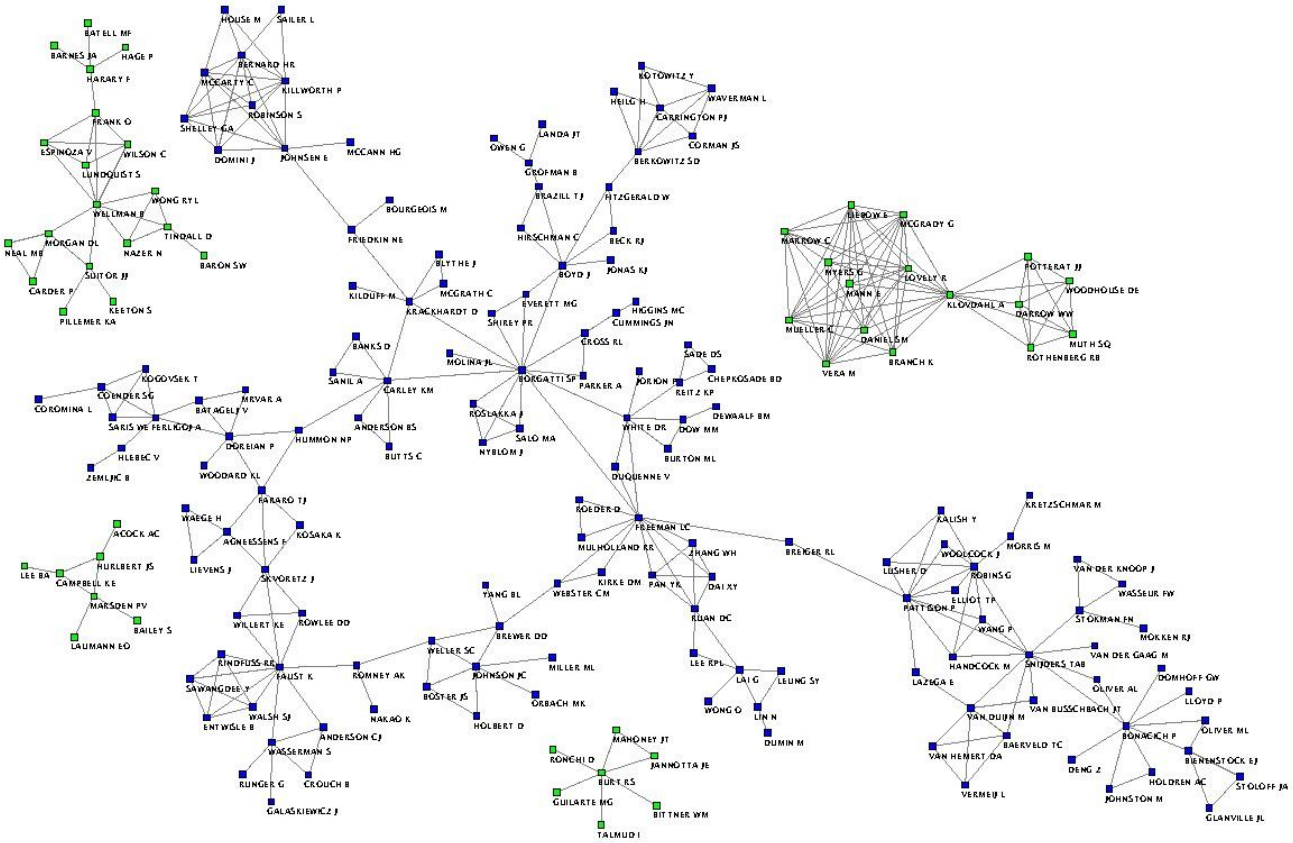

Figure 2.3: Example of Co-Authorship Map (Krebs, 2008). The nodes represent individual authors. The ones in blue form a large, well-connected community; while the ones in green are only connected within small communities.

There are several key concepts of network analysis that are fundamental for understanding and analyzing social networks. These concepts are: actors (nodes), relations (ties), degree, closeness, and betweenness.

Network data consists essentially of actors (nodes) and relations (ties). Half of the design of the network data has to do with nodes or actors that represent individuals within a network. The other half has to do with the relations or ties among all the individuals (Hanneman & Riddle, 2005). From Figure 2.3, it is visually clear that more centered actors have more ties than those at the periphery. The position of an actor often decides how

"powerful" it is: actors that are in more centered positions may have greater influence or more opportunities than the less centered ones. Therefore, the more centered actors often receive more attention and focus because they are relatively more "powerful" and potentially more influential. The term for various measures of relative "powerfulness" of an actor is called centrality.

To visualize how an actor could be "powerful" and influential, a simple structural example of a "star" network is illustrated in Figure 2.4. "Star" networks are one of the most common computer network topologies and also the main components of more complex social networks.

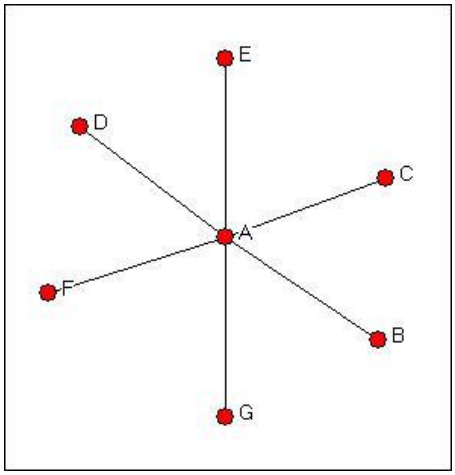

Figure 2.4: "Star" Network (Hanneman & Riddle, 2005). Each node represents an individual. Each tie represents a connection between individuals.

It is clear that actor A is in the center position in the "star" network, and it has more ties than any other actors. If A is a person, he or she has the greatest number of connections with six other people B, C, D, E, F, and G. In other words, actor A has degree of six, while all the other actors have degree of only one. Actors with more ties, such as A, are considered less dependent on other individuals since they may have access to more of

the actors within the network. Degree is therefore a simple measure of an actor's centrality.

The second reason why actor A is more "powerful" is that A is closer to more actors than any others within the "star" network. A is at a geodesic distance of one from all the other actors. However, each other actor is at a geodesic distance of two from all the other ones except for A. Actors that are able to reach other actors with shorter distances are considered to be more "powerful". The measure of how close an actor is from other actors is captured by closeness centrality.

The third reason why actor A has more advantages within the "star" network is that A lies right in between any other pairs of actors. In other words, A can directly contact any actors within the network, but if D wants to contact F, D has to go through A in order to reach F. The advantage that A has is simply being in between other actors in terms of ties. Betweenness centrality can be measured by examining which relations are most central.

Freeman (1979) developed mathematical measures of centrality for actors based on their degree, closeness and betweenness. However, since these measures are difficult and time consuming to conduct manually, software tools have been developed to conduct mathematical calculations of the measures while incorporating graphical techniques to provide a compact and systematic representation of networks. Another reason for using software tools for social network analysis is that computers are able to store and manipulate data quickly and more accurately. Efficiency and accuracy in complex data analysis are always desirable.

3.0 Methodology and Results

The goal of our project was to create a foundation for identifying which software tools work best for evaluating the impact of National Science Foundation (NSF) grants. We planned to do this using compilation, organization, analysis, and visualization software tools. These tools would enable the examination of data from sample pathways such as networks of Principal Investigators (PIs), networks of authors, and the evolution of course catalogs for colleges over time. These three pathways will serve as examples for the NSF to draw upon for future analysis. To complete these deliverables, we addressed a set of objectives. We identify those objectives below, and describe in detail the methods we used to accomplish them. The objectives were:

- Identify interactions and corresponding data which contribute to the impact of engineering education research grants after the completion of the grant
- Identify ways to collect the data that we identified as relevant
- Determine ways to organize the data
- Identify ways to analyze and visualize each type of data

3.1 Interactions and Quantifiable Data

3.1.1 Methods

To identify a set of interactions which have an influence on engineering education research, we brainstormed different pathways by which the product of an engineering education research grant would be exposed to other groups or individuals. Since we examined the output of an engineering education research grant, all of these pathways had a direct connection to the researcher or institution that was performing the research. The list of affected groups and individuals consisted of the research community, the

scientifically literate public, other institutions or universities, and students. We realized that not all of these interactions would be applicable to each engineering education grant, but we included each interaction to allow the system of software tools we outlined to be usable for many different grants.

To identify quantifiable data that exist within each category of interaction, we examined the interactions that we defined previously. We focused on data that would quantify the impact of engineering education research grants. Once we identified relevant data, we examined them and determined which data to focus on in order to complete our deliverables.

3.1.2 Results

During our analysis of the interactions that we defined, we realized the complexity of some of the data which we needed to collect in order to complete our goal. We listed many more forms of data than we actually planed on compiling and analyzing, but we feel that all of the data listed here could be useful for examining the impact of an NSF grant and may be targets for further analysis by the NSF.

Interactions between Researchers and the Research Community

In order to identify the data within the interactions between researchers and the research community, we brainstormed different links between them and examined current methods of evaluating researchers. When analyzing the interactions between the community of researchers and the individual researcher or institution performing the research, we found that sources of data included publications by the conductor of the research, conferences, and collaborative research. We determined that the data that stemmed from the publications were primarily the number of publications from an

individual or research team and the number of times those publications were cited by other researchers within the scholarly community. However, not all publications have the same prestige attached to them. For example, a publication in the journal *Science* might carry more influence than one published on the Public Library of Science (PLoS) website. In order to more adequately analyze this data, the journal that the article was published in should be taken into consideration based on article submission rejection rates and the article should be given a weight based upon the selectivity level of the journal. The data that could be gathered from conferences include the number of people who attended the conference weighted by the prestige of the conference. A more comprehensive analysis of this data type would involve taking the amount of influence the conference has into consideration. The data that came from cooperative research would reflect how many different people worked on the sponsored research grants. We evaluated this data by examining the names of the PIs and co-PIs who worked on the same grant and researchers that co-authored articles.

Interactions between Researchers and the Scientifically Literate Public

We determined through brainstorming that the scientifically literate public is connected to the researcher through public forums. The two main sources from which data could be extracted were web sites and articles in newspapers and magazines. The data within websites included the number of hits which the site received and the actual origin of the hits. The origin of the hits could help determine how many of the hits are actually by people who are interested in, and affected by, the information on the page and how many hits are from automated programs. The data from newspapers and magazines included the number of articles about the technology or information produced by the grant and the average number of readers of the newspaper or magazine.

Interactions between Researchers and Universities

In the connection between the researcher and other universities, the main sources of data are conferences and course curricula. This interaction stemmed from brainstorming along with an interview with our liaison. The data from conferences consisted of how many different universities had delegates in attendance. Further investigation of these data would lead to an investigation of which delegates attend which talks. This would allow for an analysis of which universities obtained information about the outcome of an NSF grant. The final data that we examined for this interaction were the course curricula at universities. More specifically, our target was the course descriptions located within the university catalog.

Interactions between Researchers and Student

The sources of data we identified for the interaction between the person conducting the research and a student were mainly course evaluations at some universities and evaluations of the students' performance. The course evaluations allow the student to express his or her opinion on the class and how it was taught without worrying about that opinion having any impact on their final grade, and these opinions are data which could be used. The main evaluation of a student's performance was the grades the student received. A more complete analysis of this data could be to examine individual grades of test scores to see which piece of the class the students excelled or struggled in.

Summary

Of all of the interactions we have defined so far, we will be focusing on two for the remainder of this report. These are the interaction between researchers and the research

community and interaction between researchers and universities. More specifically, the data that we found and analyzed are the collaboration data for PIs and co-PIs of awarded grants, the collaboration data for authors and co-authors of published papers, and the course curriculum data from universities as presented in course catalogs. The analysis of these types of data will most directly allow us to produce our three deliverables.

3.2 Data Compilation

3.2.1 Methods

Once we identified all of the sources of data that were useful to our project, we investigated ways to compile those data.

Interactions between Researchers and Research Community

To obtain the data on PIs and co-PIs, we decided to search the publicly available database of engineering education research grant awards on the NSF website. We chose to search for "engineering education research" because we were evaluating the impact of engineering education research grants funded by the Engineering Education and Centers (EEC) Division of the NSF. Next, we exported the information on the PIs and co-PIs who were awarded grants into an Excel spreadsheet.

To analyze the links between authors for grant-related publications, we first needed to locate the publications. We searched for these publications by examining NSF awarded grants for any indication of which publications were published because of a grant. To obtain the data for the grants, we searched the NSF's publicly available database for "engineering education research". Once we obtained the data about the grants, we

searched the data for any publications that were linked to the grant. We then identified the authors' names from the publications linked to the grants.

Interactions between Researchers and Universities

We brainstormed places where the data in the form of course descriptions could be found and determined that equivalent versions could be found in either paper or online course catalogs. We chose to look at the online course catalogues because we decided it would be the most efficient use of time, and because they are electronically accessible. We conducted a preliminary examination of the course catalogs for a variety of universities including Worcester Polytechnic Institute, Massachusetts Institute of Technology, Stanford University, California Institute of Technology, and the University of California, Berkley. We saw that the course catalogs were all either HTML or PDF files.

We then needed to search for a web crawler that could extract HTML or PDF files, or both file types. We searched on softpedia.com and google.com with the keywords "web crawler". We chose the web crawlers based on page rank, user reviews, and how the description best matched our desired functions. Once we chose the software tools that we wanted to test, we downloaded each one. We then evaluated each software tool in terms of ease of use, the ability to edit search depth, how long it takes to crawl a webpage, and the ability to filter search pages.

Ease of use is how easily the user can find and use customization options when running the program. A score of "1" in this category would mean that the program is hard to navigate and the features required to run the program, such as ability to edit search depth and filter ability, are hidden within many menus and require searching. A score of "5" would mean that the options to run the program are easy to find and require no

searching. The ability to set a search depth is the ability to set how many pages away from the root URL can be crawled. A score of "1" in this category would mean that the program had the feature to edit hidden within menus and, when the user found the feature, the program had little ability to edit the search depth. A score of "5" would mean that the feature was easy to find and the user could edit search depth easily. How long it takes to crawl a webpage is the amount of time that the program takes to complete a crawl from start to finish. A score of "1" in this category would mean that it took a few hours to crawl WPI's course catalog from the index page (http://www.wpi.edu/academics/catalogs/ugrad/sectio35.html) with a search depth of 3. A score of "5" for crawl time would mean that it took 15 seconds to crawl WPI's course catalog from the index page (http://www.wpi.edu/academics/catalogs/ugrad/sectio35.html) with a search depth of 3. The ability to filter search pages is whether or not the program has the option to filter or target certain keywords and either reject or crawl those respective web pages. A score of "1" would mean that there is little ability to create a keyword filter. A score of "5" would mean that the user can create a filter to search by certain keywords or focus on those keywords.

Once we found some promising web crawlers, we tested the software tools by using the web crawlers to extract the course information we needed from universities' course catalogs.

3.2.2 Results

We determined that the two main sources of relevant data were databases and the World Wide Web. We also found software to help collect the data on the Internet.

Interaction between Researchers and Research Community

The PI network data we found in the publicly available NSF database were already compiled. This was useful because we did not have to use a tool to compile it. One example of data we found in the database is the information about which PIs and co-PIs worked together and were assigned to investigate grants funded by the NSF. Once we found the data on PIs and co-PIs we used the core functionality of a Database Management System (DBMS) to export the data from the website into Microsoft Excel.

For the authorship network data, we successfully found the grants that applied to engineering education research from the NSF's publically available database. When we searched the information that we found for any indication of papers that were published as a result of an NSF grant, we did not find any publications. Once we determined that we could not obtain this data from the NSF's publically available database alone we asked our liaison, Alan Cheville, where the data on published papers that are linked to grants are located; Alan Cheville informed us that data on papers connected to a grant are not currently compiled for the NSF. Although the data set for our specific purpose did not yet exist, we wanted to propose a suite of software tools that could analyze a similar data set. We needed to define a data set that would closely approximate the intended data set of publications. We decided to search the ISI Web of Knowledge database of publications using the search term "engineering education research." We decided upon "engineering education research" as our search term because we assumed that many of the papers that would be found in the data set of papers connected to NSF funded engineering education research grants could also be found with that search term in ISI Web of Knowledge. To export the data to our local computer, we used an option within ISI Web of Knowledge that

allowed us to export up to 500 references at one time. We exported 500 references from ISI Web of Knowledge as a plain text file on our local computer.

Interaction between Researchers and Universities

We chose to use three different web crawling software tools to extract data from the online course catalogs of different universities. We graded them in several categories using a scale of "1" to "5" with "1" being the worst and "5" being the best. Table 3.1 (below) shows the scores each web crawler received in each category.

Table 3.1: Evaluation Scores for Web Crawlers

	Ability to Edit Search Depth	Crawl Time	Filter Ability	Ease of Use
Website Ripper Copier	5	5	5	5
Webreaper	4	4	4	4
Visual WebRipper	2	2	2	1

The first software tool we evaluated was Website Ripper Copier (Tensons Software Corporation). We found Website Ripper Copier when we searched softpedia.com for "web crawler". This tool was the second ranked software in the search. The core functionality of Website Ripper Copier is to create offline copies of both web pages and files located on those pages. Website Ripper Copier was compatible with HTML and PDF files, but excelled at extracting PDF files. We used a 30 day trial version of this software. If the user wanted to use the full version of Website Ripper Copier, the cost is $49.95 for a single license. This program allows the user to crawl websites and the files located on the websites. This program has a user friendly wizard which guides the person through the process of setting up a crawl. The settings of a crawl include the starting page, the depth of the search, and

the file types to be downloaded. The disadvantage of this wizard is that it is difficult to edit the settings of the crawl once the original crawl has been set up. We found that this program was easy to set up and, once set up, excelled at downloading files from a website. Overall, the initial setup of a crawl is simple with Website Ripper Copier, but the interface is simplified to the point where the main features we wanted were hard to find.

Webreaper (Mark Otway) is a web crawler that has the core functionality we desired. We found Webreaper when we searched for "web crawlers" on softpedia.com. Webreaper was the top downloaded web crawler on softpedia.com by roughly 15,000 downloads. The core functionality of Webreaper is to create offline copies of both web pages and files located on those pages. This program has the ability to crawl both HTML and PDF files, but excelled in extracting HTML. This web crawler allows the user to filter the crawl by the number of pages away from the original URL to limit the number of pages crawled. This minimizes the extra pages crawled. There are also other available filters, such as a keyword filter for URLs, a filter for the server that a page is on, and a filter for the format of what is downloaded. One disadvantage of Webreaper is that the layout obstructs the user from finding and utilizing the features of the software. The Webreaper program automatically stores collected data in a folder on the desktop of the user's computer, which allows easy access to the data for analysis.

Visual Web Ripper (Sequentum) is another web crawler that we evaluated. When we searched google.com with the keywords "web crawler", Visual Web Ripper was the second available software with a free download. The homepage said that Visual Web Ripper is able to extract complete data structures, such as catalogs (Sequentum, 2010). This program was not good at extracting either HTML or PDF. Visual Web Ripper allows the user to either input one URL, or a list of URLs as starting points. Once the URL is

specified, the page is loaded into Visual Web Ripper. The user must then highlight what he or she wants to keep from each webpage. A major disadvantage of this program is the interface. For example, the function to export the collected data is extremely hard to find and is not automated. Also, Visual Web Ripper did not automatically crawl from page to page, but only went where the user instructed the program to go. This program has many features which are good for the collection of specific data from specific pages, but is not suited for the crawling and downloading of many linked pages.

In our analysis of these web crawling software tools, we crawled the online course catalogs of a few different schools. There were several factors that affected the ease of crawling the catalogs. Worcester Polytechnic Institute's (WPI) course catalog was easy to crawl because it had an HTML format and a singular index to crawl from. However, the course catalogs for some universities were difficult to crawl. Oregon State University's catalog was difficult to crawl because it had 6841 classes available. This means that we would need to search 6841 pages, which is quite tedious because there is no singular index page to crawl from. University of Connecticut's undergraduate catalog was in a PDF format, which was easily crawled with Website Ripper Copier. In order to crawl these websites, we needed to use different web crawlers which are formatted with HTML and PDF inputs.

In our attempts to crawl course catalogs on different university's websites, we found that Stanford had an especially difficult catalog to crawl. All the courses were in web pages that were in a poor layout for web crawling. We could search for courses but we could not find a link from any course page to another. There were over 6000 pages with classes on each page. A web crawler cannot complete a single crawl using any of these URLs as a seed

page. So, if someone wanted to crawl Stanford's entire course catalog, they would have to do over 6000 web crawls, which would take too much time to be feasible.

WPI had an easy website to crawl because of its layout. The website had a link to WPI's academic departments. From each department's page in the catalog, we could access the descriptions of courses offered in that department because they were all listed on that webpage. Each page containing the course descriptions were a maximum of 3 pages away from the seed page. This was ideal for the use of a web crawler because the user could set a search depth of 3 so that a "crawl" could be completed for the entire catalog with one URL.

3.3 Data Organization

3.3.1 Methods

Once we compiled the data, we needed to organize the data into a form which was easily usable. This consisted of organizing the files where the data was stored as well as standardizing the format of all of the data collected.

Interactions between Researchers and Research Community

For interactions within the research community, we examined the EEC PI and co-PI network and a sample authorship network. We first examined the data for the PI network. We considered whether to split the data into separate files using different software or to keep it compiled in one Microsoft Excel file. We considered data storage and ease of access to the data when making this decision. We determined that keeping the data in a single Microsoft Excel file would be the most feasible method of organization for this data. We then considered how to filter and organize the data within the Microsoft Excel file. We examined different ways to isolate the names of the PIs and co-PIs and to convert all the

names into one format. The pathway that we examined only used functions built into Microsoft Excel because we did not find the need to explore a different software tool.

We then examined the sample authorship network. We realized that the final format of the data we wanted to extract from the text file was the same as the format for the PI network. We concluded that the process of conversion and organization of collected data would be similar to that of the conversion and organization process we used for the PI network once the data was in an Excel spreadsheet. We then thought about the different ways to convert the plain text file into an Excel file. Analogous to the PI data, we gave preference to functions built into Excel because we did not find it necessary to add another program to the suite of tools.

Interactions between Researchers and Universities

We began by deciding that it would be easier to format all the collected data uniformly. We then brainstormed and compared formats which would be viable options for an output format. The most viable format options were PDF, HTML, and text files. We chose the formats of PDF and HTML for consideration because we preferred to have to convert only a subset of the collected data. The disadvantage of these formats was that there was a lot of text formatting included in the files which made it harder to process data from the files. Text files contain only text with minimal formatting, so these files can easily be opened by any program which reads text. We chose to work with text files because they are recognized by any text reading program and contain just the raw data that we wanted.

Once we determined the target file type, we began the search for tools which would convert HTML and PDF files to text files using softpedia.com and google.com. We searched for tools that would convert both file types as well as tools that would convert only one of

the file types. We decided to use the exact function of the tool and to search for the theory behind the tool in the search terms. We did this so that we could get a sample of results that were more focused on one file type as well a set of results that encompassed a broader idea. An example of searching for the exact function of the tool which we performed is a search for "HTML to text converter" or "PDF to text converter". An example of searching for the theory behind the tool which we performed is a search for "Text mining tool". We selected the tools to analyze from the search results based upon their rating by other users, their rank in the search results, and our initial impressions upon reading the program description.

Once we selected the tools to analyze, we ranked the tools on their ability to convert multiple files, ease of implementation, cost, and the quality of output. When evaluating the ability to convert multiple files, a score of "5" would mean that the software converted multiple files while only asking for the list of targets and where to place them once, and a score of "1" would indicate that there was no way to convert multiple files without running the program on each target individually. When examining the ease of implementation, a score of "5" would mean that the program was intuitive and less than 5 minutes were required to learn the basic features of the program, and a score of "1" would indicate that learning the basic functionalities of the program took more than one hour. When evaluating the quality of the output, a score of "5" would indicate that the output contained all of the text we wanted in a layout identical to the target file as well as having no extra formatting or text, and a score of "1" would indicate that the output did not follow the layout of the target file, there was missing or additional text, and there was additional formatting.

Once the files were converted to the text format, we had to determine how to store the files. We decided that the storage system did not have to be complicated and that the system simply needed to provide an intuitive system of organization for the converted text files. We searched for systems we could use to organize this data in the manner we wanted.

3.3.2 Results

For data organization, we found software tools to organize the PI and authorship network data, and we also found software tools to organize the files obtained from using web crawlers.

Interaction between Researchers and Research Community

The easiest way to isolate the relevant data from the EEC PI and co-PI network was to put filters on the columns of the excel file. We used these filters to show only the grants that were given out under the EEC division. Once we filtered out the excess data, we hid all of the columns accept for the PI and Co-PI columns. We then encountered a problem with the format of the names. Some of the names were last name first and some were first name first. We noticed that, when the last name was first, there was always a comma between the names, and when the first name was first, there was no punctuation between them, only a random number of spaces. To change all of the names to first name first, we converted the name column into two columns by using the comma as the trigger to split the data. We then combined the data back together but in reverse order to organize the name category as first name first.

The other issues that we faced was that the co-PI column was divided up by vertical bars and the first and last names were separated by a random number of spaces. Similar to the PI names, we split the data into separate columns using the vertical bars. We then

replaced all of the spaces between the names with a single space in order to make all of the names uniform. The result of this process was an organized spreadsheet where the names sharing a row were connected by a project. A sample of this data is shown below.

Table 3.2: Sample Data of EEC Awardees

	PI	Co-PI			
Award #1	Stephanie Adams	Rosalyn Hobson			
Award #2	Ahmet Aktan	Thomas Hewett	Patrick Gurian	Franklin Moon	Franco Montalto
Award #3	Cynthia Atman	Deborah Kilgore			

To transfer the data for the authorship network from the text file to an Excel format, we used the import data function embedded within Excel. This allowed us to perform a similar style of conversion as for the PI data, detailed above. One major difference between the two sets of data was that, in the PI data, the co-PI column was split by vertical bars, while in the sample authorship data, the authors were all listed below one another in the same column. This resulted in two problems: how to identify the authors that worked together and how to organize the names of authors who worked on one publication on one line. The first problem was solved by examining the surrounding data. There was always a line that had the title of the article before the authors. We used this standardization to identify which authors collaborated on a publication. Once we knew which authors worked on publications together, we needed to figure out how to get the names of the authors who worked together onto one line in the spreadsheet. This was solved with the transpose function that is programmed into Excel. We copied all of the authors who worked together from the column and pasted each group of authors into its own row by using the transpose option under paste special. Once we did this, we uniformly formatted the names using the

same process we used for the PI network. When this was done, the authorship data was organized in the same way as the PI data.

Interaction between Researchers and Universities

The tools that we selected from our searches were Text Mining Tool 1.1.42 (text-mining-tool.com), Detagger (JafSoft), and Some PDF to Word Converter (SomePDF.com) and Zilla PDF to TXT Converter (PDF Zilla). We used the data which we previously collected to test this software. We graded the software tools in several categories using a scale of "1" to "5" with "1" being the worst and "5" being the best. Table 3.3 (below) shows the scores each file conversion tool received.

Table 3.3: Evaluation Scores for File Conversion Software

Software Name	Ability to convert multiple files	Ease of implementation	Quality of output
Text Mining Tool 1.1.42	3	5	5
Detagger	4	4	4
Some PDF to Word Converter	5	4	4
Zilla PDF to TXT Converter	5	4	2

One of the tools we used to convert the data we pulled from web sites to text is called Text Mining Tool 1.1.42. This tool came in two forms, a menu-driven version and a command line version, both of which are free software. This freeware's main functionality is to convert an HTML file into a text file with no formatting. An advantage of the menu driven program is that it has the ability to show a preview of the text while allowing the user to either save the text to another file or copy the text straight to the clipboard. The command line version automatically saves the text as an external file to a location the user

defines. The major advantage of the command line version is that it can be used to convert many files at once. The disadvantage to both versions of this program is that the program can only be used to convert HTML files to text and cannot handle any other input files.

Another tool we used to convert data to text is called Detagger. In order to evaluate this program, we used a 30 day trial version of it; the full version costs $29.95 for a single license. The function of this program was to either convert a HTML file into a text file or remove the markups within the HTML code. There is a command line version of this software, but the menu-driven application was more user friendly and could convert many files at once. This guided our decision to give less consideration to the command line version of the software. The trial version of the program places a header at the top of every output file and randomly changes the case of some letters. These are issues which would be solved by purchasing the software.

Some PDF to Word Converter is a free conversion tool which converts PDF files into word files. The program is free and is menu-driven software. The software converts a PDF file into a text file by either utilizing text boxes or no formatting at all. If the output file is saved as a plain text file with no formatting, it is easier to read. During conversion, the lines of the text are sometimes broken up in a strange manner. Some of the time this leads to a confusing arrangement of text in the output file.

Zilla PDF to TXT Converter 1.0 is a conversion tool that converts PDF files into plain text files. This converter offers a free trial, which only would convert the first page of the PDF file. Once the user purchases the software for $19.95, he or she will be able to convert the entire PDF file. The output is a plain text file, which has no formatting compared to the PDF file. For example, the lines were not broken up in an organized

fashion and the user had to search around the new plain text file to find where the information from the PDF was placed.

When searching for simple file organization and storage systems, we found that a simple file structure on a computer desktop would suit our needs. We created one folder entitled "course catalogs" and, within that file, we made a file for each school from which we pulled data. The course catalog information was stored within the respective school's file. This system worked perfectly for one data pull but, once we performed the data search another time, we had to find a way to distinguish between the two data sets. To solve this issue, we further divided the school folders into folders labeled with the date the data was pulled and then inserted the data into the correct folder.

3.4 Data Analysis and Visualization

3.4.1 Methods

Once the data were organized, we needed to find software tools to analyze the data and produce visuals to help the user interpret the analysis.

Interactions between Researcher and Research Community

In order to determine which data analysis method is the most suitable for studying interactions between individuals, we spoke with our liaison, Alan Cheville. From our discussion, we determined that social network analysis would be the most appropriate method. Since social network analysis focuses on linking individuals through their interdependencies on other individuals, it would provide a way to examine how, and how well, the individual researchers are connected to the rest of the research community.

Our strategy was to use software programs to analyze the links between PIs and co-PIs who have received award funds from the EEC. This strategy allowed us to view which PIs or co-PIs have worked together on any EEC funded research, and how well each PI and co-PI is connected to the rest of the EEC PI community.

We started the process of choosing which social network analysis software programs can best serve our purpose by searching through google.com with keywords "social network analysis software". We found a table of Collection of Social Network Analysis Tools and Libraries on Wikipedia, which is essentially a list of social network analysis software tools. This collection summarizes the main functionalities, license and cost, input and output formats, and platforms for each software program (Wikipedia, 2010). It was helpful to us in determining which software tools we would be able to download and test. After browsing websites and user manuals of individual software programs, we chose to test one that is compatible with the NSF PI data format (Microsoft Excel spreadsheet), flexible in manipulating imported data, and capable of graphically representing networks based on various measures of centrality.

In addition to searching for social network analysis software programs based on the information available online, we also spoke with evaluation experts at the Science and Technology Policy Institute (STPI) to see which software program they have been using to analyze networks. The conversation with these experts gave us an idea of which functionalities of social network analysis software programs are most important for evaluating the impact of engineering education research grants.

We tested and compared each software program based on three qualities: ease of use in modifying and manipulating imported data, graphical functionality, and overall user-

friendliness. We then evaluated each software program on a scale of "1" to "5" for each of the qualities. For ease of use in modifying and manipulating imported data, a "1" indicates that there is a lot of difficulty in doing so, and a "5" means that it is easy to accomplish. For graphical functionality, a "1" means that it is difficult to present a network graph based on different measures of centrality, and a "5" means that a network graph can be clearly presented based on different measures of centrality. Finally, for overall user-friendliness, a "1" indicates that the software is hard to learn to use efficiently and the generated network graphs are difficult to understand; while a "5" indicates that the software is easy to operate, the user can navigate the menus intuitively, and the generated network graphs are easy for the user with some general knowledge of social network analysis to interpret.

Interactions between Researcher and Universities

One type of analysis and visualization tool we utilized for the interactions between a researcher and universities was text comparison software, which allows the user to compare two documents. We chose text comparison software tools because we decided that they would be the most efficient type of tool to use for comparing almost identical sets of text. We wanted a software tool that would do this because we wanted to monitor the changes in individual course descriptions at an institution after the implementation of the product of an engineering education research grant. This allowed us to gauge the impact of the grant through examining course descriptions because we could assess whether the engineering education grant made an impact on instruction at the university level. We were also able to track these changes over time. The text comparison software tools we analyzed were found through a web search on google.com with the search term "text comparison tool". We chose our text comparison tools because they initially appeared to be

the most useful for our purposes based on user reviews and the description of the program provided by the developer.

We tested and compared the text comparison tools based on ease of use, quality of produced visualization of the analysis, customizability, and ease of saving analyses for later use. We gave each software tool a score for each category of evaluation based on a scale of "1" to "5", with "1" being the worst score and "5" being the best. A score of "1" in the category of ease of use would mean that the software is not easy to use for someone unfamiliar with similar software tools; a score of "5" in the same category means that someone unfamiliar with similar tools would be able to sit and use the tool with little or no problem. A score of "1" in the category of quality of produced visualization of the analysis would mean that the text comparison tool did not clearly show where the differences between the documents were; a score of "5" in the same category means that the differences were all clearly marked and easy to find. A score of "1" in the category of customizability means that there were no options to customize the analysis; a score of "5" in the same category means that there were many options available to customize the analysis. A score of "1" in the category of ease of saving would mean that the software tool had no option to save generated analyses; a score of "5" would mean that there were options available to easily save the analysis and visualization for later use. When testing each text comparison software tool, we used a data file that we found on a university web site, using a web crawler, and then converted to a plain text file, using a file converter, as a standard collection of words to be analyzed. We compared the original text file to a copy of the same file in which we inserted and deleted random words for the software tools to find.

Another analysis tool that is useful for the interactions between the researcher and universities is word (or tag) clouds. Word clouds are visualizations that emphasize the

relative frequency of a word within a sample of text by the font size of the word within the visualization. Some word cloud generators remove common words, such as "if", from the word clouds so that the relative proportions of the words are not distorted. The produced visualization is usually aesthetically pleasing, but the only particularly important aspects of the visual are the relative font size and how well common words were removed from the image. Analysis of the generated visual allows the user to easily infer the main topic of the original text through the most commonly used words of the original text. We decided to test word clouds because we determined that they could be used to analyze the changes in an entire curriculum, which could include syllabi or sets of course descriptions, over any interval of time for which we could gather data. The word cloud generating tools we found and compared were found using a web search on google.com with the search term "word cloud generators". We chose the word cloud generators based on their rank in the search results and our initial impressions upon working with the tools.

We compared the word cloud generators based on ease of use, quality of produced visualization of the analysis, customizability, and ease of saving analyses for later use. We gave each software tool a score for each category of evaluation based on a scale of "1" to "5", with "1" being the worst score and "5" being the best. A score of "1" in the category of ease of use would mean that the software is not easy to use for someone unfamiliar with similar software tools; a score of "5" in the same category means that someone unfamiliar with similar tools would be able to sit and use the tool with little or no problem. A score of "1" in the category of quality of produced visualization of the analysis would mean that the word cloud generator did not remove common words from the cloud and did not combine words with different capitalization, such as "text" and "Text"; a score of "5" in the same category means that the word cloud generator accurately removed common words from the cloud and

was able to recognize words with different capitalizations as the same word. A score of "1" in the category of customizability means that there were no options to customize the analysis or visualization; a score of "5" in the same category means that there were many options available to customize the analysis or visualization. Customization of the analysis involves being able to specify aspects of the analysis such as the number of words in the cloud or the number of times a word has to appear in the text in order to appear in the cloud. Customization of the visualization involves being able to change aesthetic aspects such as color scheme or orientation of text within a word clouds. A score of "1" in the category of ease of saving would mean that the word cloud generator had no option to easily save generated analyses; a score of "5" would mean that there were options available to easily save all analyses and visualizations for later use. When testing each text comparison software tool, we used a data file that we found on a university web site, using a web crawler, and then converted to a plain text file, using a file converter, as a standard collection of words to be analyzed.

3.4.2 Results

For data analysis and visualization, we found software tools to analyze and visualize the connections among the PIs and co-PIs, and authors and co-authors. We also found software tools to analyze course descriptions.

Interaction between Researchers and Research Community

We tested two social network analysis software programs: UCINET (Analytic Technologies) and Gephi (Association Gephi). We graded the software tools in several categories using a scale of "1" to "5" with "1" being the worst and "5" being the best. The following table (Table 3.4) summarizes the evaluation scores that each social network analysis software program received based on the functionalities we tested:

Table 3.4: Evaluation Scores for Social Network Analysis Software

	Ease of Use in Modifying and Manipulating Imported Data	Graphical Functionalities	Overall User Friendliness
UCINET	4	3	3
Gephi	5	5	5

The first social network analysis software program we tested was called UCINET. We acquired the name of this software program from the professional network analysts of the STPI.

UCINET is a menu-driven Windows software program. Using the 60-day limited free-trial version of the software, we explored the main functions of UCINET and tested these functions with actual network data. The cost of the program varies by type of customer after the free-trail period. For US government agencies, such as the NSF, the program costs $150. UCINET can import data files in formats such as Excel spreadsheet and text files, describe all data as matrices, and output graphs of social networks.

The imported data files need to be converted into a matrix because network data are generally presented in a square data array, also called a square matrix. As mentioned earlier (Section 2.4), the purpose of using a square matrix is to be able to focus on presenting actors (nodes) and the relationships among them. For example, the network example given in Table 3.2 can be presented in a square matrix shown in the following table (Table 3.5).

Table 3.5: Square Matrix of the Sample Data of EEC Awardees (shown in Table 3.2)
A "1" demonstrates a connection between the pair of PIs or co-PIs; a "0" demonstrates no connection between the pair.

	Stephanie Adams	Rosalyn Hobson	Ahmet Aktan	Thomas Hewett	Patrick Gurian	Franklin Moon	Franco Montalto	Cynthia Atman	Deborah Kilgore
Stephanie Adams	0	1	0	0	0	0	0	0	0
Rosalyn Hobson	1	0	0	0	0	0	0	0	0
Ahmet Aktan	0	0	0	1	1	1	1	0	0
Thomas Hewett	0	0	1	0	0	0	0	0	0
Patrick Gurian	0	0	1	0	0	0	0	0	0
Franklin Moon	0	0	1	0	0	0	0	0	0
Franco Montalto	0	0	1	0	0	0	0	0	0
Cynthia Atman	0	0	0	0	0	0	0	0	1
Deborah Kilgore	0	0	0	0	0	0	0	1	0

A "1" in a cell means that there is a connection between the pair of PIs whose row or column contains that cell. Similarly a "0" represents no connection between the pair. However, this matrix can get extremely large when the network is more complex due to more awards and PIs. Instead of having to insert each binary measure of connection manually, UCINET can generate such a large matrix using the imported data file. This is a great function because it makes data processing much less time consuming and prevents errors arising from inputting entries manually.

UCINET graphs social networks using a program named NetDraw, which is integrated with UCINET. NetDraw is able to take any data matrix generated by UCINET and produce a map of the networks established within the matrix. We imported the network data of the EEC PI community organized into the format shown in Table 3.2 to test UCINET. The graph that was generated by NetDraw is shown in Figure 3.1.

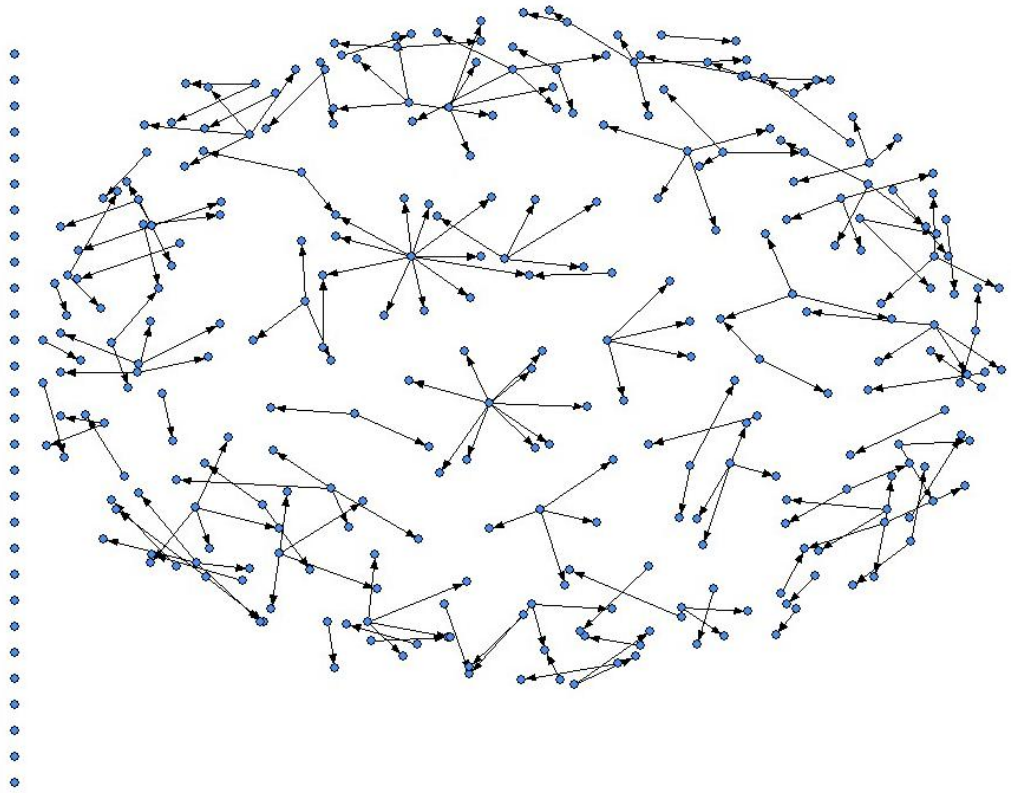

Figure 3.1: Graph of the EEC PI and Co-PI Network (without Names), UCINET and NetDraw. Each node represents either a PI or a co-PI who was awarded grant(s) by the EEC between 2003 and 2010. The nodes with arrows pointing outward are PIs, and the ones with arrows pointing inward are co-PIs. The list of nodes on the left consists of the PIs who have not worked with anyone else within this network. The location of each node was randomly assigned.

In Figure 3.1, each blue node represents either a PI or co-PI. The ones with arrows pointing outward are PIs, and, similarly, the ones with arrows pointing towards them are co-PIs. The connected nodes represent PIs and co-PIs who have worked together. The list of nodes on the left consists of the PIs who have not worked with anyone else within this network.

UCINET is also capable of graphically presenting various measures of centrality. An example of the EEC PI community measured by degree centrality is shown in Figure

3.2. This feature is extremely helpful when visually analyzing which PIs are best connected within the network.

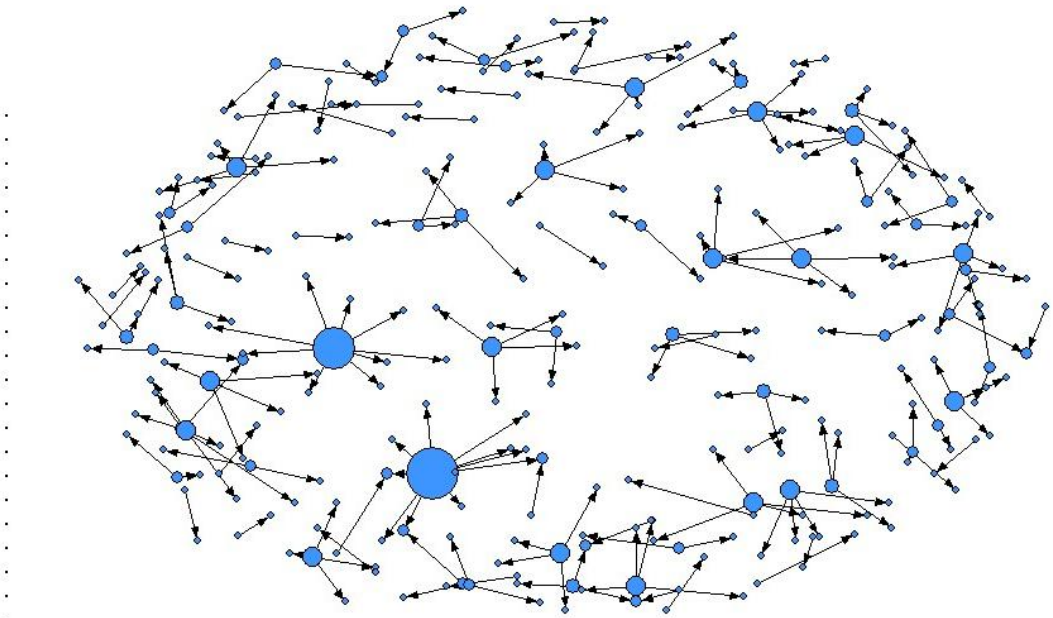

Figure 3.2: Graph of the EEC PI and Co-PI Network (without Names) Based on Degree Centrality, UCINET and NetDraw. Each node represents either a PI or a co-PI who was awarded grant(s) by the EEC between 2003 and 2010. The nodes with arrows pointing outward are PIs, and the ones with arrows pointing inward are co-PIs. The list of nodes on the left consists of the PIs who have not worked with anyone else within this network. The location of each node was randomly assigned. The size of a node corresponds to its degree centrality. The more connections a node has, the bigger the node becomes.

Figure 3.3 (below) shows the same network as in Figure 3.2 with the PI names. Looking at the figure, one may think that this graph looks over-populated with nodes, arrows, and especially the names of the PIs. Since the labels of names cannot be generated

with a transparent background, nor is it possible to relocate them, there is no way to avoid overlapping of labels, nodes, and arrows.

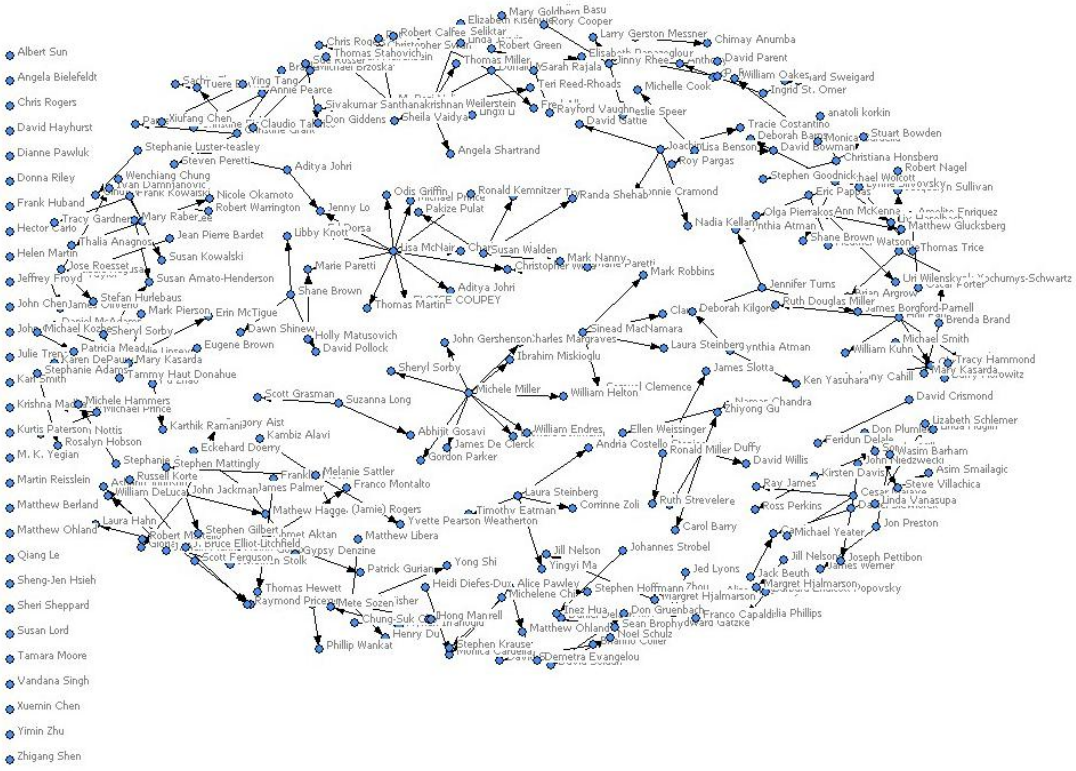

Figure 3.3: Graph of the EEC PI and Co-PI Network (with Names), UCINET and NetDraw. Each node represents either a PI or a co-PI who was awarded grant(s) by the EEC between 2003 and 2010. The nodes with arrows pointing outward are PIs, and the ones with arrows pointing inward are co-PIs. The list of nodes on the left consists of the PIs who have not worked with anyone else within this network. The location of each node was randomly assigned.

As clearly shown earlier in Figure 3.1, the graph looks much cleaner without all the PI names. However, the names of the PIs and co-PIs are essential for the analysis of the network connections. Having to go through the program menu to turn the labels on and off is inconvenient.

Another desired function we were looking for was a function to color code the nodes based on different attributes, such as the gender of the PIs and Co-PIs. The "Node Attribute Editor" is a function in NetDraw that allowed us to color code by adding new attributes to individual nodes. The resulting graph color coded by gender is shown in Figure 3.4, where the males are presented in black and the females are presented in red.

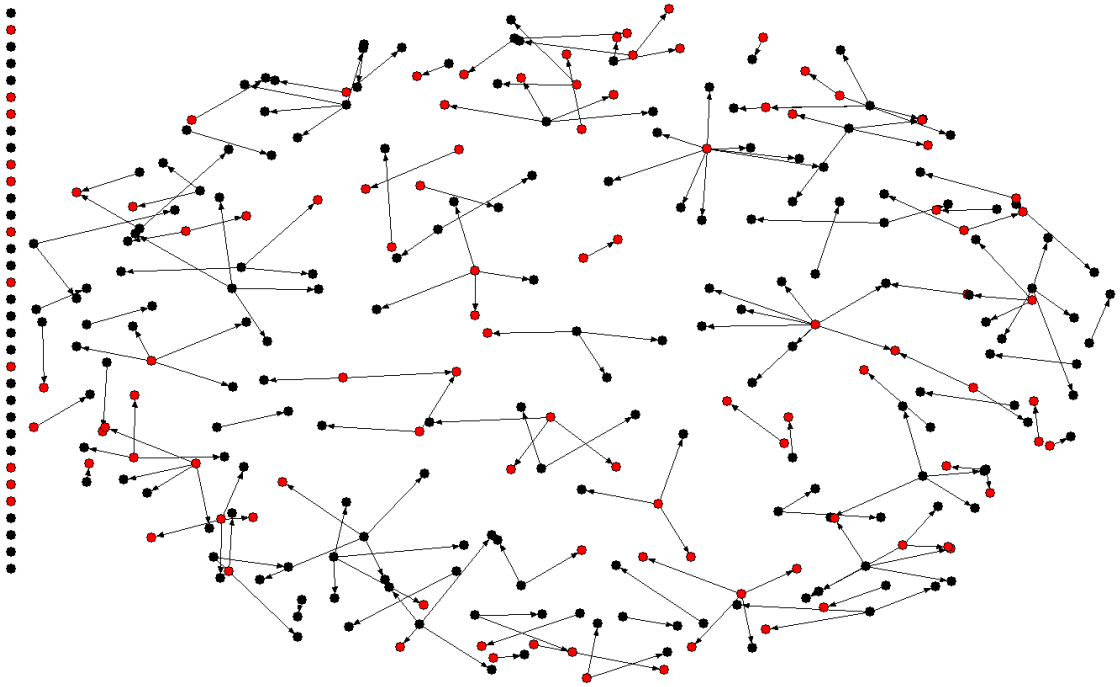

Figure 3.4: Graph of the EEC PI and Co-PI Network (without Names) (Color Coded by Gender: Male-Black, Female-Red), UCINET and NetDraw. Each node represents either a PI or a co-PI who was awarded grant(s) by the EEC between 2003 and 2010. The nodes with arrows pointing outward are PIs, and the ones with arrows pointing inward are co-PIs. The list of nodes on the left consists of the PIs who have not worked with anyone else within this network. The location of each node was randomly assigned.

We were also able to import the data from the sample authorship network. We applied similar procedures to this data using UCINET to get the following graph (Figure 3.5) of this network. The graph looks similar to the one of the EEC PI and co-PI network,

except this one has more nodes because there were more actors in our sample authorship data than there were in the EEC PI and co-PI network.

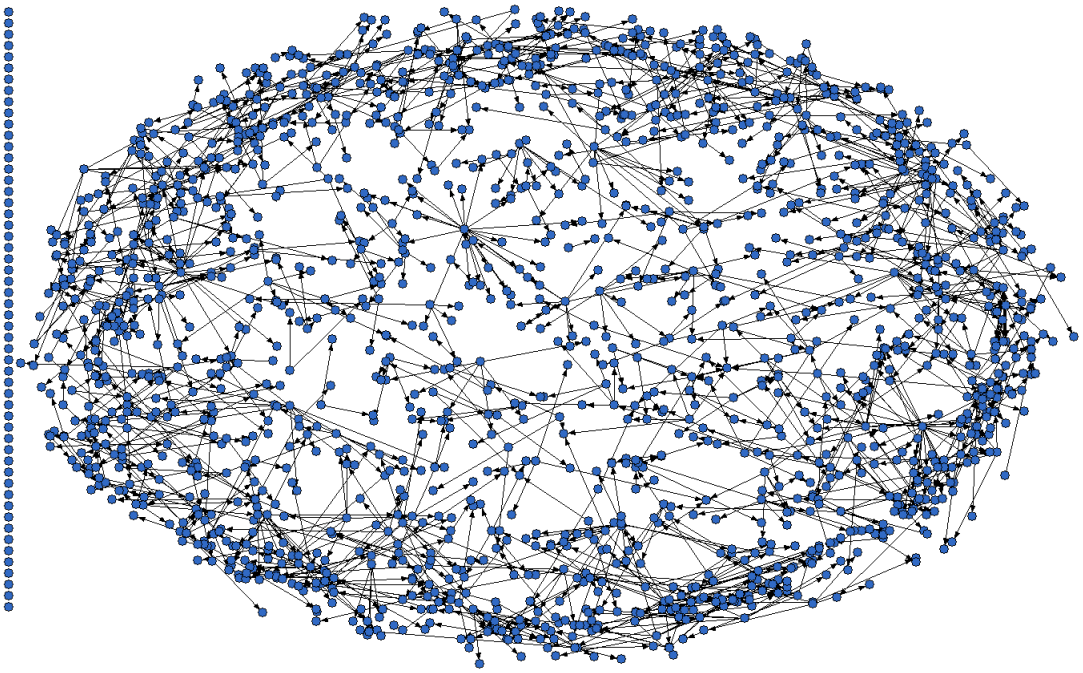

Figure 3.5: Graph of the Sample Authorship Network (without Names), UCINET and NetDraw. Each node represents either an author or a co-author within the sample authorship data we compiled. The nodes with arrows pointing outward are authors, and the ones with arrows pointing inward are co-authors. The list of nodes on the left consists of the authors who have not worked with anyone else within this network on publications within the chosen field. The location of each node was randomly assigned.

The second software program we tested was called Gephi. We acquired the name of this software from the Collection of Social Network Analysis Tools and Libraries on Wikipedia.

Gephi is an open-source network visualization tool compatible with Windows, Mac OX, and Linux. Gephi was created "with the idea to be the Photoshop of network visualization" (Gephi, 2010). Similar to UCINET, it can read data files formatted in

Comma Separated Values (CSV) spreadsheet, generate matrices, and produce graphs of the networks described within the matrices. However, the most noticeable features of Gephi are its rich functionalities around the visualization window that allow the user to analyze and visualize a network in various ways. In addition to graphically showing measures of centrality by altering the size of the nodes, Gephi is also capable of showing measures of centrality by changing the color density of the nodes. If the user combines these features, a PI or co-PI who is well connected would be represented by a node that is bigger and has a darker color.

Another useful feature of Gephi is that it shows the weight of ties automatically. For example, if PI A has only worked with PI B once, there would be only one connection between PI A and PI B, and this connection has the weight of "1"; however, if PI A has worked with PI B twice, the connection between them is weighted "2". Gephi counts the number of connections among all the nodes from the data file and connects those who have more connections with a bolder line.

We used Gephi to generate a graph of the EEC PI and co-PI network. As shown in Figure 3.6, there are no single nodes present, because nodes that do not have any connections were automatically neglected. The two nodes with the largest size and the darkest color represent the two PIs who have the highest degree centrality, or more connections within the network. PIs and co-PIs that have lower degree centrality are shown with smaller nodes and with lighter colors. It is quite intuitive to tell which connections are weighted more than the others are by simply looking at the boldness of them. The labels of names were taken off to provide a cleaner graphical example, but there are various options for viewing the labels in the visualization window. The visualization

window also allows interactions, such as emphasizing a node and its immediate neighbors if the user hovers the cursor over that node by making the other nodes fade.

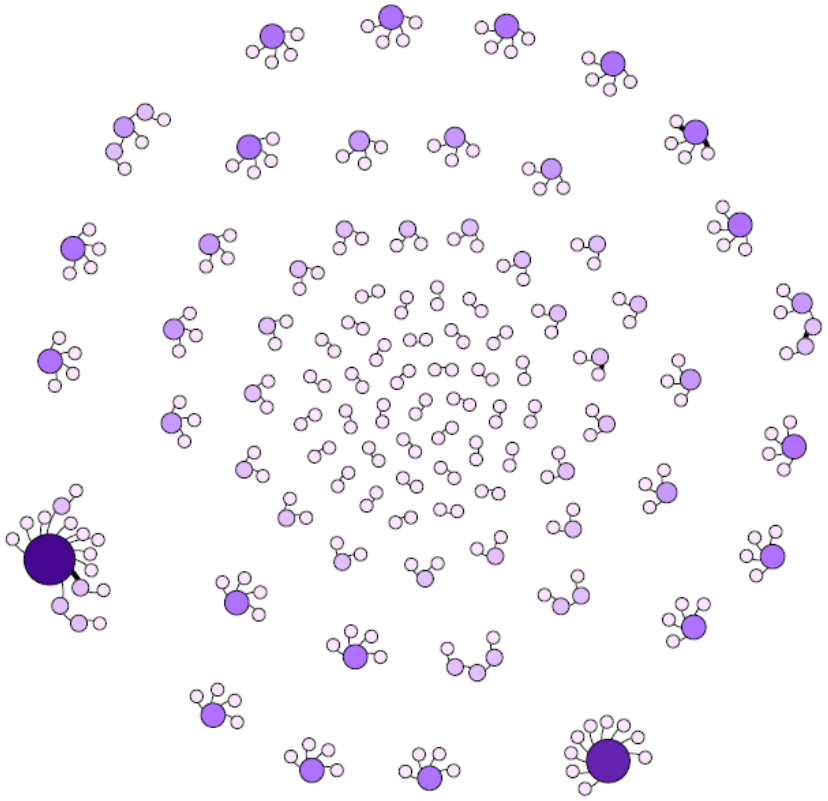

Figure 3.6: Graph of the EEC PI and Co-PI Network (without Names), Gephi. Each node represents either a PI or a co-PI who was awarded grants by the EEC between 2003 and 2010. The size and color density of a node correspond to its degree centrality. The more connections a node has, the larger in shape and darker in color it becomes. The arrows pointing from PIs to co-PIs are not visible at this zoom level.

With regard to data, Gephi organizes imported network data in the "Data Laboratory" which presents the entire list of the nodes and allows the user to add new attributes as well as modify properties of individual nodes. We inserted gender information to each of the node within the EEC PI and co-PI network, and configured the males to be

presented in blue and females to be presented in pink by changing the color property. Figure 3.7 below shows the resulting graph color coded by gender.

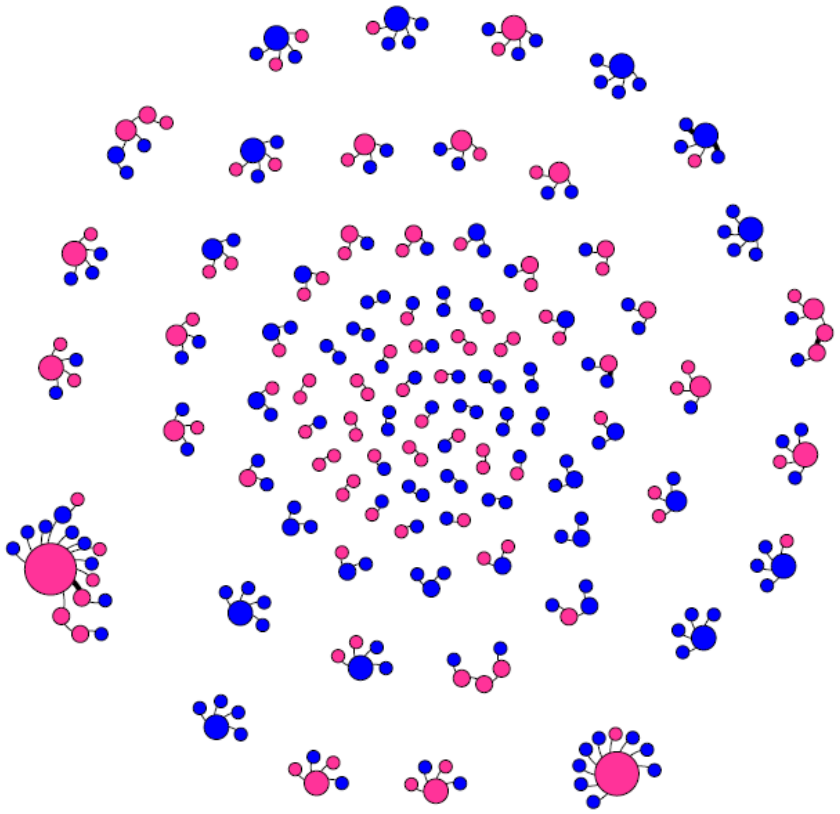

Figure 3.7: Graph of Network of the EEC PI and Co-PI Network (without Names) (Color Coded by Gender: Male-Blue, Female-Pink), Gephi. Each node represents either a PI or a co-PI who was awarded grants by the EEC between 2003 and 2010. The size of a node corresponds to its degree centrality. The more connections a node has, the bigger in shape it becomes. The arrows pointing out from PIs to co-PIs are not visible at this zoom level.

We applied similar procedures with the data from the sample authorship network. The following figure (Figure 3.8) illustrates the graph of this network. The node positions were randomly distributed. The degree centrality of the nodes is shown with different sizes and color densities.

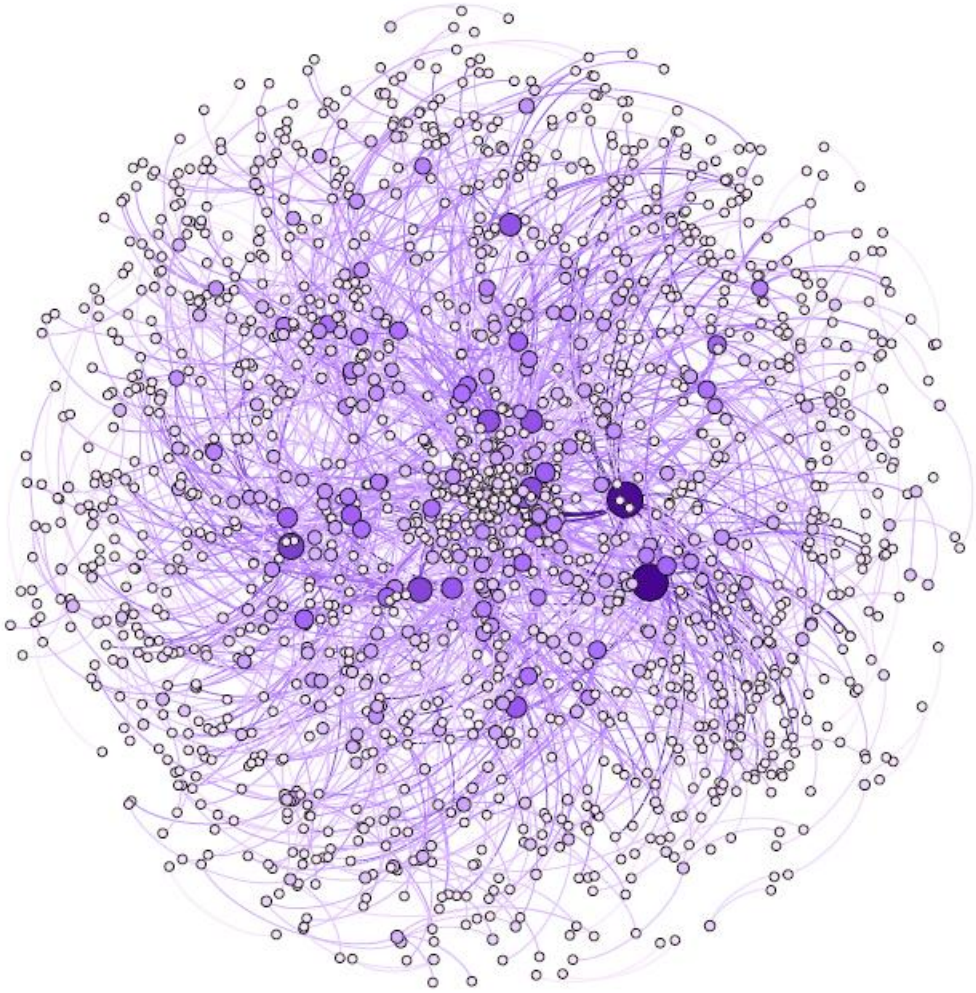

Figure 3.8: Graph of the Sample Authorship Network (without Names), Gephi. Each node represents either an author or a co-author within the sample authorship data we compiled. The arrows pointing out from authors to co-authors are not visible at this zoom level. The location of each node was randomly assigned.

Interaction between Researchers and Universities

The three text comparison software tools we decided to test were KDiff3 (Joachim Eibl), ExamDiff (PrestoSoft), and WinMerge (WinMerge Development Team). We graded the software tools in several categories using a scale of "1" to "5" with "1" being the worst

and “5” being the best. The following table (Table 3.6) shows the score each software tool received in each category of evaluation.

Table 3.6: Evaluation Scores for Text Comparison Tools

Software Tool	Ease of Use	Quality of Visual	Customizability	Options for Saving
KDiff3	4	5	4	4
ExamDiff	4	4	4	4
WinMerge	4	3	4	4

The first text comparison software, KDiff3, requires a simple file upload to view the documents in the program window. KDiff3 displays two documents side by side and has the default feature to highlight differences between the two documents. This feature highlights the line that contains the change in one color and the actual change in a lighter version of the same color. There is also an easily accessible function that allows the user to change between documents to compare, and another function that allows the user to merge documents together. Furthermore, the program allows the user to show white space as colored underscores. This emphasizes any changes that may have been made to formatting.

The second text comparison software, ExamDiff, also requires a simple file upload to view the documents in the program window and displays two documents beside each other. Visualization of differences between documents is easy with ExamDiff; added lines that are not present in the other document different are red, changed lines are displayed in yellow, and deleted lines are in blue. There is also an option to highlight changes and to display nothing more than those changed lines. The changes are stored in a drop-down menu that allows the user to easily select the changes between the documents.

The third text comparison software, WinMerge, also requires a simple file upload to view the documents in the program window and displays two documents beside each other. WinMerge allows the user to have more than one text comparison project open at once, in different tabs. The user can alternate between the tabs when they want to see a different project. Visualization of differences between documents is easy with WinMerge because the lines of a document that are different are highlighted and the locations of the differences within the line are highlighted in a lighter shade of the same color. The program allows the user to scroll through the differences between the documents and contains a function that allows the user to merge documents together.

The four word cloud generators we decided to test were Tagxedo (Hardy Leung), Wordle (Jonathan Feinberg), TagCrowd (Daniel Steinbock), and TagCloudMaker (calevans). We graded the software tools in several categories using a scale of "1" to "5" with "1" being the worst and "5" being the best. The following table (Table 3.7) shows the score each word cloud generator received in each category of evaluation.

Table 3.7: Evaluation Scores for Word Cloud Generators

Software Tool	Ease of Use	Quality of Visual	Customizability	Options for Saving
Tagxedo	3	5	4	4
Wordle	4	4	4	1
TagCrowd	4	5	2	3
TagCloudMaker	2	3	1	1

The first word cloud generator we analyzed, Tagxedo, is a web-based word cloud generator and is fairly easy to use. The user can submit text to Tagxedo in multiple ways. The user can upload a text file, copy the desired text and paste the text into a box in the submission portion of the web page, or, if the text is on another web page, type in a web

address of a page that you want the program to analyze. The quality of the generated visualization is good, and the program has a large number of customization options for the image. However, we thought that the options to customize of the image were not quite intuitive and required some experimenting with buttons to figure out. Tagxedo offers a wide variety of formats and qualities for saving the word clouds generated. Below (Figure 3.9) is an example of a word cloud generated by Tagxedo from a text file of the chemical engineering courses in the WPI 2010-2011 course catalog.

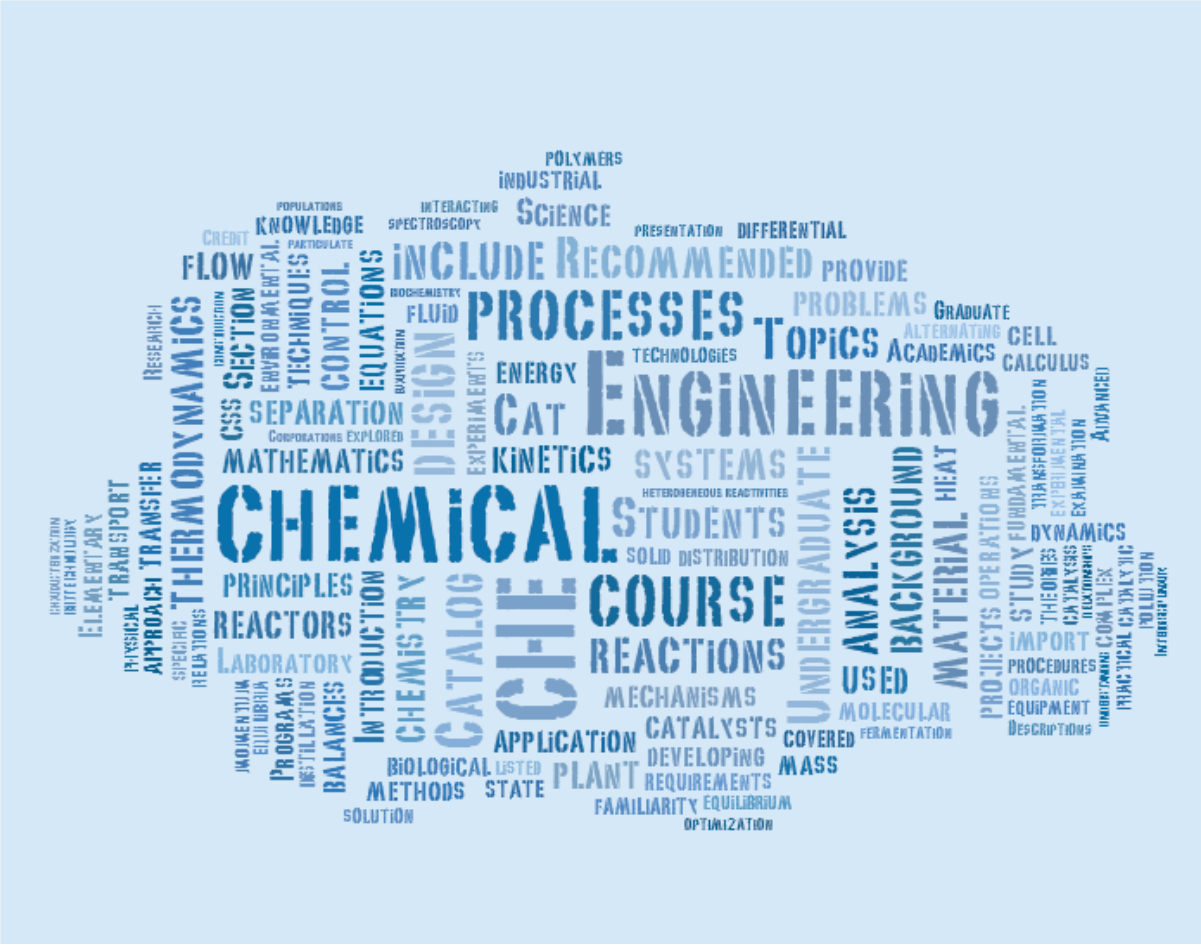

Figure 3.9: Word Cloud of WPI Chemical Engineering Courses Generated by Tagxedo. The only aspect of this visualization that is essential to the analysis of the document is the relative size of the words. This means that the words in the largest font size appear within the text most frequently and the font size and word frequency decrease correspondingly. The color, orientation, and relative proximity to other words have no analytical meaning.

The second word cloud generator to be analyzed, Wordle, is also a web-based word cloud generator and easy to use. The user can submit text to Wordle in two ways: he or she can copy the desired text and paste the text into a box in the submission portion of the web page, or, if the text is on another web page, type in a web address of a page that the user wants the program to analyze. The generated image is aesthetically pleasing, and the program has a large number of customization options for the image. However, Wordle does not have the capability to recognize a word with different capitalization, such as "chemical" versus "Chemical", as the same word. This impacts the quality of the produced visualization because it distorts the scale of the font size of the words. Wordle allows the user to print their visualization or save it to a public gallery. Below (Figure 3.10) is an example of a word cloud generated by Wordle from a text file of the chemical engineering courses in the WPI 2010-2011 course catalog.

Figure 3.10: Word Cloud of WPI Chemical Engineering Courses Generated by Wordle. The only aspect of this visualization that is essential to the analysis of the document is the relative size of the words. This means that the words in the largest font size appear within the text most frequently and the font size and word frequency decrease correspondingly. The color, orientation, and relative proximity to other words have no analytical meaning.

The third word cloud generator to be analyzed, TagCrowd, is a web-based word cloud generator that is still in beta form and is easy to use. The user can submit text to TagCrowd in multiple ways. The user can upload a plain text file, copy the desired text and paste the text into a box in the submission portion of the web page, or, if the text is on another web page, type in a web address of a page that the user wants the program to analyze. The quality of the generated image is good because the image does not include common words such as "the" and combines words despite different capitalizations, but the program does not have options for customization of the image. It does, however, allow the

user to customize options such as the number of words that appear in the word cloud or how many times a word should appear in the text before it can be included in the cloud. TagCrowd allows the user to print their visualization or save it as a PDF or HTML embed. Below (Figure 3.11) is an example of a word cloud generated by TagCrowd from a text file of the chemical engineering courses in the WPI 2010-2011 course catalog.

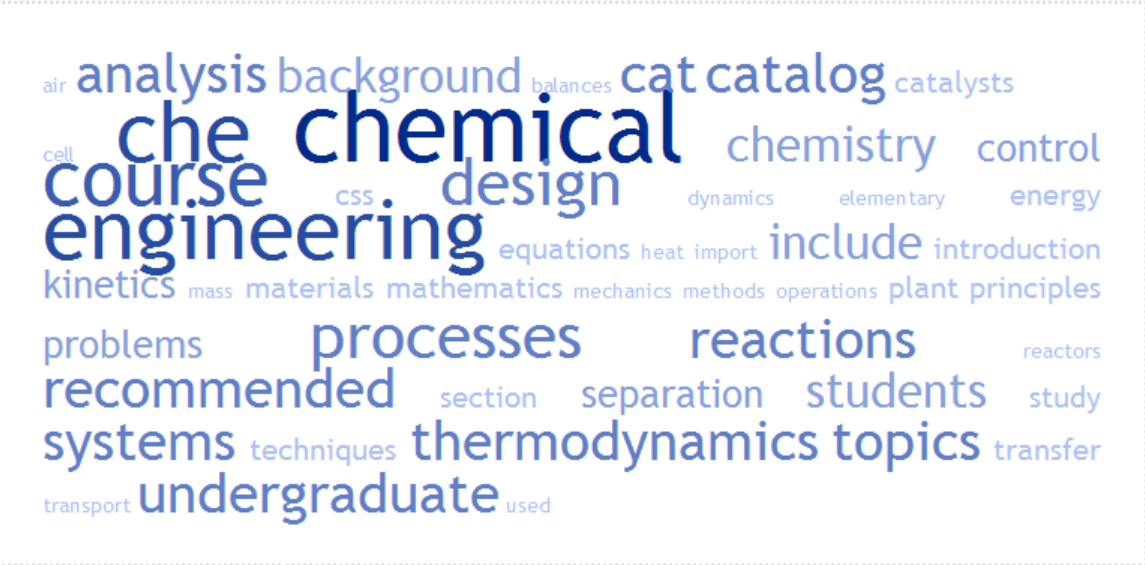

Figure 3.11: Word Cloud of WPI Chemical Engineering Courses Generated by TagCrowd. The only aspect of this visualization that is essential to the analysis of the document is the relative size of the words. This means that the words in the largest font size appear within the text most frequently and the font size and word frequency decrease correspondingly. The color has no analytical meaning, and this word cloud generator just happens to organize the words within the cloud alphabetically.

The fourth word cloud generator to be analyzed, TagCloudMaker, is a downloadable word cloud generator that is relatively easy to use, but requires some familiarity with similar software tools. To use TagCloudMaker, the user first needs to tell it which folder to get the text from. The program must be directed to a folder with plain text files in it; TagCloudMaker does not allow the user to select individual files. The user then instructs TagCloudMaker which folder to save the generated image to. There is an option to exclude

a list of commonly used words that is included with the download and is customizable, but this functionality is not refined and the program may still include the words on the list in the word cloud. This decreases the quality of the visualization because common words, such as "the", take emphasis away from the more important words in the word cloud. The program has no options for the customization of the image. TagCloudMaker can save a local copy of the image, but if another image is saved to the same folder TagCloudMaker will overwrite the existing file with the new file. Below (Figure 3.12) is an example of a word cloud generated by TagCloudMaker from a plain text file of the chemical engineering courses in the WPI 2010-2011 course catalog.

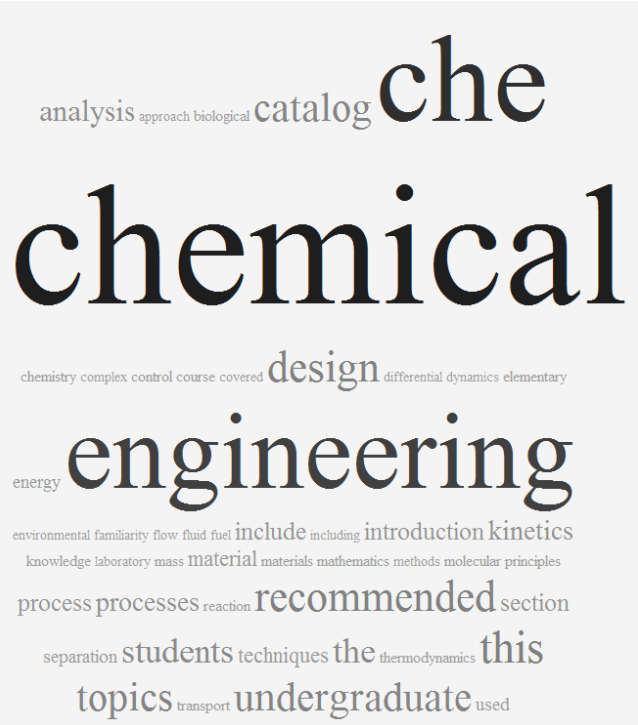

Figure 3.12: Word Cloud of WPI Chemical Engineering Courses Generated by TagCloudMaker. The only aspect of this visualization that is essential to the analysis of the document is the relative size of the words. This means that the words in the largest font size appear within the text most frequently and the font size and word frequency decrease correspondingly. The color and relative proximity to other words have no analytical meaning.

4.0 Discussion

We have individually compiled and analyzed a list of software tools that would enable us to complete our overall goal of outlining a suite of software tools which will most effectively allow the National Science Foundation (NSF) to encourage researchers and institutions to have the greatest impact on the field of engineering education. Now we will compare the software tools in each individual category against the others in that category to ascertain which software we believe to be the best to use for each step of each process.

4.1 Data Compilation

We analyzed the data compilation software tools on the categories of the ability to edit the search depth, the crawl time, filter ability, and the ease of use. Once we analyzed all of the software tools in each category, we compared the tools against each other to determine the best data compilation software tool for each deliverable.

4.1.1 Interactions between Researchers and Research Community

Since all of the data for the Division of Engineering Education and Centers (EEC) Principal Investigator (PI) and co-Principal Investigator (co-PI) network and authorship network that we needed to compile was contained within publicly available databases, we only had to utilize the software used to interface with these databases. For our purposes, the search and export ability through the NSF web site provided all of the functionality that we needed for the EEC PI and co-PI network along with the convenience of not having to add another program to our suite of tools. For the authorship network, we utilized the same abilities through ISI Web of Knowledge's export function.

4.1.2 Interactions between Researchers and Universities

When we tested the three web crawlers, Website Ripper Copier, Webreaper, and Visual Web Ripper, we found that each program had advantages and disadvantages. We compared them on their ability to edit search depth, crawl time, ability to filter, and ease of use. The first category we compared the software tools on was their ability to edit search depth. Website Ripper Copier had the ability to edit the search depth while the links were being uploaded; that option was immediately apparent. Webreaper's ability to edit search depth required searching within the program, which was inconvenient when compared to Website Ripper Copier's more apparent feature. Visual Web Ripper was much less intuitive and required searching through multiple menus to find the option to edit search depth.

We then compared the crawl time of the software tools. Website Ripper Copier was able to complete its crawl in mere seconds, whereas Webreaper took a couple minutes to crawl after the URL was entered. Visual Web Ripper took a significantly longer time to select all the web pages we wanted to crawl and format the web pages within multiple menus.

We then compared the web crawlers based on filter ability. Filter ability was built into the beginning menu for Website Ripper Copier, but Webreaper required a small amount of searching to set a filter depth. Visual Web Ripper required excessive searching of menus to set a filter, which, compared to the other two, was quite extensive. Website Ripper Copier had an introductory menu with everything needed to execute a crawl, including filter depth editing, filter ability, and the ability to enter the URL for the seed page, making everything quite easy. Webreaper allowed the user to easily enter the URL of the seed page. Webreaper's layout was fairly easy to navigate; the layout enabled the user to easily find the options for filter ability and search depth. Visual Web Ripper was not as

user friendly as Webreaper. The menus were misleading when we were trying to find the options to edit filters and how deep we wanted Visual Web Ripper to crawl. Visual Web Ripper was also difficult to use because we had to manually select the web page we wanted to crawl.

4.2 Data Organization

We analyzed the data organization software on the categories of the ability convert multiple files, the ease of implementation, and the quality of the output. Once we analyzed all of the data organization software in each category, we compared them against each other to determine the best data organization software for each deliverable.

4.2.1 Interactions between Researchers and Research Community

Since all of the changes that we had to make to the data were contained within one Microsoft Excel file, we did not find the need to use any tool besides Microsoft Excel. Microsoft Excel provided all the functionality we needed along with the convenience of not having to add another program to the suite of tools.

4.2.2 Interactions between Researchers and Universities

First we will examine the two software tools that convert HTML files to text files. The three main categories on which we analyzed the data conversion software were the ability to convert multiple files, the ease of implementation, and the quality of the output. Of these three categories, we decided that the most important category was the quality of the output. This is because we needed the outputs to be as close to the original data as possible, while retaining a high ease of access to other programs such as text comparison tools. Text Mining Tool 1.1.42 had the highest score in this category for multiple reasons.

The main reason was that, when using the same HTML file as an input for both Text Mining Tool and Detagger, Text Mining Tool had the raw text data that we wanted and no extra formatting.

The next important category was the ease of implementation. Text Mining Tool had a command line version which facilitates the conversion of files with one simple command line argument. When looking at the Graphical User Interface (GUI) for both Detagger and Text Mining Tool, there is no distinct advantage or disadvantage to either one. Text Mining Tool has the better ease of implementation because of its ability to take advantage of batch files and the command prompt to automate file conversion.

The final category on which we evaluated the software tools was the ability to convert multiple files. Both of the software tools have this ability, but they differ in the way in which they achieve this. Text Mining Tool can only convert multiple files through the utilization of the command line version. Detagger has the ability to select multiple files and convert them without the need to change the target of multiple lines of code within the command prompt. Detagger has the better score in this category because of its ability to convert multiple files with a faster setup time. Although Detagger received a higher score in this category, the other two categories were determined to be more important. Therefore, Text Mining Tool is the better choice for a HTML to text conversion tool.

Some PDF to Word Converter and PDF to TXT Converter were the tools we examined for the conversion of PDF files to text files. The importance of the categories in which these conversion tools were compared was the same as when evaluating the HTML to text software tools. Therefore, the first category we compared the PDF to text file converters was the quality of the output. Some PDF to Word Converter had a clear

advantage in this category. Some PDF to Word Converter was able to extract the text and place it on the page in the same general locations, whereas PDF to TXT Converter did not preserve the page layout during conversion.

The next category by which we evaluated the software was the ease of implementation. The software tools both had a similar graphical user interface and therefore a similar ease of implementation. Both of the tools had the buttons necessary to select and convert files easily displayed for the user to find and use.

The final category by which we evaluated the software was the ability to convert multiple files. Similar to the ease of implementation, the tools were alike in this category. Both of the software tools made it easy for the user to select multiple files to convert. From our analysis, we concluded that the PDF conversion tools were similar and differed only in the quality of the output. Some PDF to Word Converter was the best choice for a PDF to text conversion tool because of its higher output quality.

4.3 Data Analysis and Visualization

We examined two categories of data analysis and visualization software. For social network analysis software, the categories were the ease of use in modifying and manipulating imported data, graphical functionalities, and overall user friendliness. For text analysis software and word cloud generators, the categories were ease of use, quality of visual, customizability, and options for saving. Once we analyzed all of the software on these categories individually, we compared the social network analysis software as well as text comparison software and word cloud generators to determine the best data analysis and visualization software for each deliverable.

4.3.1 Interactions between Researchers and Research Community

By comparing UCINET with Gephi, we were able to conclude that both software tools have similar basic functions and require similar operational procedures. However, according to Table 3.4, Gephi generally received higher evaluation scores than UCINET based on the functionalities we tested. For ease of use in modifying and manipulating imported network data, the "Data Laboratory" in Gephi is a user-friendly interface for modifying data such as adding attributes to the nodes and changing graphical property of individual nodes. In contrast, UCINET does not have a data modification interface which is as centralized. As for graphical functionalities, even though both Gephi and UCINET have the basic functions to produce network graphs based on common measures of centrality, Gephi clearly generates graphs that are more intuitive to read, and easier to modify according to the user's need. Furthermore, since Gephi is an open-source software program, it allows users and developers to suggest new functions to add on and to extend functionalities by uploading program codes to the Gephi Plugin Portal with ease. This feature allows third party organizations, such as the NSF, to customize the software program easily to best meet their needs. Thus, Gephi received a higher score in "Overall User Friendliness".

4.3.2 Interactions between Researchers and Universities

The text comparison tools were ultimately similar for each of the qualities by which we compared them. There was little difference between the ease of use, customizability, and options for saving for each of the software tools. This means that the quality of the visualization became the distinguishing characteristic between the software tools. Since the quality of the visual produced by KDiff3 was better than the visuals produced by

ExamDiff and WinMerge, we decided that KDiff3 is the most preferable text comparison software tool of the three we tested.

Each word cloud generation tool has advantages and disadvantages. The biggest disadvantages of TagCloudMaker are that it is less user friendly than the web based tools, it overwrites its own saved files, and it does not completely remove common words, such as "the", from the word cloud (see Figure 3.12 for an example). The biggest advantage of TagCloudMaker is the ease with which the user can submit more than one document to be analyzed in the same word cloud, since there would just have to be more than one document in the selected folder, to find the most common words within a group of documents. This is a useful function because word clouds would be best suited to help the user examine the overall curriculum of a course of study, and can be used to determine the major changes in the curriculum if data is compiled over time.

All three web based tools we evaluated were much more user friendly than TagCloudMaker and are much better about removing common words from the visualization, but do not allow the user to upload more than one document to the software, if the document upload option is available at all. This inability to upload more than one document at a time can easily be circumvented by simply copying and pasting the text from more than one document into the appropriate box in each respective web based word cloud generator. Since there is a simple way to negate this disadvantage of web based tools, the only advantage to TagCloudMaker is not much of an advantage. This resulted in a group consensus that the web based word cloud generators we evaluated were all better than TagCloudMaker.

Of the three web-based word cloud generators we evaluated, the quality of the visualization varies little between TagCrowd and Tagxedo. The quality of the visualization generated by Wordle is diminished by its inability to recognize a word with different capitalization, such as "chemical" versus "Chemical" (see Figure 3.10 for an example), as the same word. Since these words were counted separately, the relative scale of the words in the cloud was thrown out of proportion. The ability to save generated images using the software was also inconsistent between the web based tools. Tagxedo had many different options for saving the generated image, TagCrowd had a couple of different saving options, and Wordle had one option besides printing. Since saving the generated images is essential to comparing word clouds, we thought that the available saving options were an important measure by which the word cloud generators should be evaluated. The quality of the visualizations produced and the available options for saving visualizations in Tagxedo and TagCrowd far surpassed those of Wordle.

The two remaining word cloud generators, Tagxedo and TagCrowd, have far different customization options. TagCrowd allows the user to customize few aspects of the visualization, and none of these aspects include the color scheme or orientation of text within the word cloud. Since the ability to create an image that the user feels is visually pleasing is important for visualization tools, Tagxedo is the better option between these two word cloud generators. Overall, Tagxedo is the best word cloud generator of the four we analyzed based on our defined criteria.

The documents with which we tested these tools are generally publically available, so there are no privacy concerns surrounding the use of word clouds for this purpose. However, we considered the possibility that the NSF may at some point wish to use word clouds to analyze documents that are not publically available. In response to this

realization, we developed a procedure for making the document unreadable while keeping the words within the document intact. This procedure is detailed in Appendix D.

4.4 Extension to Other Data Types

Even though we tested the software tools using specific types of data for specific purposes, the process of data compilation, data organization, and data analysis and visualization using these software tools can be applied to a broad variety of data types. In Section 3.1, we identified a set of interactions and corresponding data that contribute to the impact of engineering education research grants. However, due to time constraints, we decided to focus on only two of these interactions for the remainder of this report: interactions between researchers and the research community, and interactions between researchers and universities. Although we did not focus on any of the other interactions, we determined that the software tools could be applied to other data types within all of the interactions.

Examples of other types of data that the suite of software tools we outlined could be applied to include, but are not limited to: faculty Curriculum Vitae (CV), annual and final reports submitted for grants, university course syllabi, publication abstracts, and conference attendees. A CV could be examined to see how an NSF grant has impacted the professional life of a teacher. Analysis of annual and final reports submitted for grants can determine whether or not the PI met the expectations of the NSF for a particular grant. University course syllabi can be examined to determine if an NSF grant has had a positive impact on a course. Publication abstracts could be analyzed for keywords to easily visualize the main idea of the paper, which could then show how a grant has impacted different

fields. Analysis of the network formed by conference attendees could provide an idea of where the knowledge of the outcome of an NSF grant could spread to.

5.0 Recommendations

Through the course of this project, we have noticed some areas that could be expanded or improved upon in the future to allow the National Science Foundation (NSF) to perform a more complete analysis of the impact of its grants.

5.1 Social Network Analysis Software

Since our project was constrained by the time limit of seven weeks, our testing of the social network analysis software programs was limited. We were unable to test and compare every function of UCINET and Gephi because there are simply too many functions provided by these powerful software tools to enable us to fully judge them in the time we had. Therefore, **we recommend that the Division of Engineering Education and Centers (EEC) further explore the network analysis functions of both of these software programs as well as create tutorials for both social network analysis tools**.

Tutorials can be helpful for teaching interested staff members about how to use the software programs. Even though UCINET has a comprehensive set of tutorials available online, it would take a while for a first time user to figure out how the instructions and concepts apply to the specific data they are analyzing. The tutorials for Gephi are not as detailed, so it requires the user to explore and learn the functionality of Gephi by themselves. **We recommend the NSF develop in-depth tutorials to help streamline the learning process for a new user.**

We recommend that NSF investigate tools that can be used to analyze the changes in a network over time. This tool would be used in conjunction with social network analysis tools to visualize the evolution of networks.

5.2 Format of Data

When we were organizing the EEC Principal Investigator (PI) and co-Principal Investigator (co-PI) data that we used to test the social network analysis tools, we found that it would have been easier if the format of the data was standardized. The PI names were organized in the format of [last name, first name] with a comma in between last and first names, while the co-PI names were in the format of [first name, last name] with a random number of spaces in between first and last names. The fact that we had to reformat the names using functions in Excel was not time efficient. **We recommend that the NSF have a uniform format for similar data entries**, so that the data are ready to be imported to software tools, such as social network analysis tools, as needed.

5.3 University Website Structure

When using a web crawler for online course catalogs, we ran into the issue of not being able to use one tool to crawl the different catalogs. This problem was created by the use of both HTML and PDF files in the catalogs and varying site structure for the different universities. The first issue could be solved by developing a web crawler that could crawl course catalogs for both PDF files and HTML files, distinguish between those file types, and only download the appropriate files. This would reduce the amount of time necessary to search through the downloaded data and find the applicable files. This would also reduce the amount of hard drive space required to store the offline files. The second issue is the

more complex of the two and would require the web crawler to have a way to determine what the site structure is and how to crawl it.

We recommend that a web crawler be developed that would be able to start at a university's home page and be able to locate and download just the data that the user wants, not the whole web page. This would require that the web crawler be able to analyze keywords provided by the user and compare the given keywords to keywords located in links identifying the web pages. This would allow the user to use a web crawler for the collection of course catalogs stored in different site structures, faculty Curriculum Vitae (CV), project information, or any other data stored on a university's web page.

5.4 Publication Data

When searching for publication data for the authorship network, we had to define and compile our own data set because the data of publications that were connected to an NSF grant was not already compiled. **We recommend that the NSF require that a PI reports any publications that result from the grant to the NSF.** This would enable the NSF to analyze the authorship data for these publications using a similar process to the one we defined for the authorship data.

6.0 Conclusions

The ultimate goal of this project was to facilitate evaluating the impact of engineering education research grants funded by the Division of Engineering Education and Centers (EEC) of the National Science Foundation (NSF) through the use of software tools. We accomplished this by creating plans for effective procedures to use for the analysis of a network of Principal Investigators (PI) and co-Principal Investigators (co-PI), a network of authors and co-authors, and the evolution of college course catalogs over time. For the analyses, we created a procedure which included data collection, data organization, and data analysis and visualization.

We found that the data for PIs and co-PIs are accessible from the publicly available NSF grant database. Thus, the data collection phase for the analysis of networks of PIs and co-PIs is relatively simple. The user simply has to access the database and export the names of the PIs and co-PIs. The organization phase is more user intensive, because the names have to be formatted uniformly. Microsoft Excel is the most convenient tool to use for organizing these data. Once the format of the data is correct, the user can import the data to a social network analysis tool to visually analyze the network. We recommend using Gephi for the network analysis and visualization.

For the authorship network, we decided that a sample of authors within one field would be sufficient to provide an example of how to prepare an analysis of a similar network. We found that we could export information such as abbreviated author names, full author names, and titles for published papers from an online database. We exported a sample size of 500 publications using the keywords “engineering education research” from the database to analyze. We then manually extracted the authorship data and organized

them in a Microsoft Excel spreadsheet. We extracted the data manually because there is currently no software tool that will complete this task for us. Once the data were organized in a uniform format, the user could import the data to a social network analysis tool to visually analyze the network. We recommend using Gephi for the network analysis and visualization.

We decided that the most useful data for the analysis of the evolution of course catalogs over time would be course descriptions. We then decided that the best web crawler for collecting course descriptions in PDF was Website Ripper Copier. However, Webreaper was the best web crawler for extracting course data in HTML. Once the data are collected, they must be organized and converted into a format that can be easily analyzed. The best way to organize the collected course descriptions is through a series of file folders on the user's computer. The best software to convert an HTML file to plain text was Text Mining Tool and the best software to convert a PDF file was Some PDF to Word Converter. Once the text is in the desired format, it can be analyzed for different purposes. The two main purposes for analysis we considered were the analysis of individual course descriptions and the analysis of the overall curriculum of a program. When analyzing the individual course descriptions, we decided that the best type of software tool would be text comparison tools, and ExamDiff is the best choice among this type. When analyzing the overall curriculum of a program, the best type of software tool would be word cloud generators. The best word cloud generator we evaluated was Tagxedo.

The NSF will be able to use web crawlers, Microsoft Excel, text conversion tools, text comparison tools, word cloud generators, and social network analysis software tools for a variety of data types. For example, a web crawler can be used to compile any data available on the internet. Microsoft Excel can be used to organize any network data. Text conversion

tools can convert HTML and PDF files to plain text tiles, and there are probably tools available that will convert other text formats to plain text files. Text comparison tools can be used to analyze any pair of nearly identical plain text files. Word cloud generators can be used for any block of text that needs analysis. Social network analysis tools can be used to analyze any networks and quantify the interactions within those networks.

Bibliography

Balakrishnan, A., Lal, B., & Flattau, P. E. (2008). *The Evaluation of Engineering Education Research: Emerging Issues and Promising Developments.* Washington: Institute for Defense Analyses Science & Technology Policy Institute.

Frechtling, J. (2002). *The 2002 User Friendly Handbook for Project Evaluation .* Washington, DC: National Science Foundation.

Freeman, L. C. (1979). Centrality in Social Networks Conceptual Comparison. *Social Networks* , 215-239.

Freeman, L. C. (2004). *The Development of Social Network Analysis: A Study in the Sociology of Science.* Vancouver: Empirical Press.

Friedman, V. (2008, January 14). *Data Visualization and Infographics*. Retrieved November 10, 2010, from Smashing Magazine: http://www.smashingmagazine.com/2008/01/14/monday-inspiration-data-visualization-and-infographics

Friendly, M., & Denis, D. (2001). *Milestones in the history of thematic cartography, statistical graphics, and data visualization.* Retrieved November 9, 2010, from Department of Mathematics and Statistics: York University: http://www.math.yorku.ca/SCS/Gallery/milestone/

Gephi. (2010, July 15). *Manyfesto - Gephi*. Retrieved November 17, 2010, from Gephi:Wiki: http://wiki.gephi.org/index.php/Manifesto

Hanneman, R. A., & Riddle, M. (2005). *Introduction to Social Network Methods.* Riverside: University of California, Riverside.

Kim, P. (2010, November 1). Is Higher Education Evolving? *Educause Quarterly* .

Knoke, D., & Yang, S. (2008). *Social Network Analysis.* Thousand Oaks: Sage Publications, Inc.

Krebs, V. (2008). *Coauthorship Map of Social Network Scholars*. Retrieved November 10, 2010, from Orgnet.com: http://www.orgnet.com/SN.html

Lal, B. (2009). *Strategies for Evaluating Engineering Education Research.* Washington: Institute for Defense Analyses Science & Technology Policy Institute.

National Science Foundation. (2001). *FY 2002 GPRA Performance Plan.* National Science Foundation.

National Science Foundation. (2010, April 7). *NSF at a Glance*. Retrieved September 19, 2010, from National Science Foundation Web Site: http://www.nsf.gov/about/glance.jsp

Pant, G., Srinivasan, P., & Menczer, F. (2004). Crawling the Web. In M. Levene, & A. Poulovassilis, *Web Dynamics: Adapting to Change in Content, Size, Topology and Use* (pp. 153-178). Springer.

Science and Technology Policy Institute. (2009). *Welcome*. Retrieved October 3, 2010, from Science and Technology Policy Institute Web Site: https://www.ida.org/stpi/

Sequentum. (2010). *Visual Web Ripper*. Retrieved December 1, 2010, from Visual Web Ripper Overview: http://www.visualwebripper.com/ProductInformation/Overview.aspx

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2001). *Database System Concepts.* McGraw-Hill Science Engineering.

Statistics Canada. (1998). *Statistics Canada Quality Guide.* Ottawa: Minister of Industry.

Stufflebeam, D. L., & Shinkfield, A. J. (2007). *Evaluation Theory, Models & Applications.* San Francisco: Jossey-Bass.

Wikipedia. (2010, November 3). *Social Network Analysis Software*. Retrieved November 15, 2010, from Wikipedia: http://en.wikipedia.org/wiki/Social_network_analysis_software#cite_note-23

Appendix A: The National Science Foundation

The National Science Foundation (NSF) (2010) was created in 1950 as an independent federal agency. The mission of the NSF is to "promote the progress of science; to advance the national health, prosperity, and welfare; to secure the national defense" (NSF at a Glance). The NSF carries out their mission by funding approximately 20 percent of all the basic research done at colleges and universities throughout the United States with an annual budget of about $6.9 billion (FY 2010). They fulfill their mission by awarding about 10,000 grants per year which last about 3 years each, to fund the most promising research proposals. "Most of these awards go to individuals or small groups of investigators. Others provide funding for research centers, instruments and facilities that allow scientists, engineers and students to work at the outermost frontiers of knowledge" (NSF at a Glance). Through awarding these grants, NSF continues to discover where the brightest horizons of science lie.

NSF (2010) consists of seven major directorates that support science and engineering research and education: Biological Sciences, Computer and Information Science and Engineering, Engineering, Geosciences, Mathematics and Physical Sciences, Social, Behavioral and Economic Sciences, and Education and Human Resources, as shown in Figure A.1. Each directorate has an assistant director and each is divided into several divisions. There are nine other offices that also support research and researchers. NSF leadership is mainly comprised of a director and a National Science Board (NSB). The NSB consists of 24 members who are appointed by the President of the United States. The director of the board is also appointed by the President of the United States and confirmed

by the United States Senate. There are roughly 2100 NSF employees in Arlington, V.A., including 1400 career employees, 200 temporary scientists, and 450 contract workers (Who We Are).

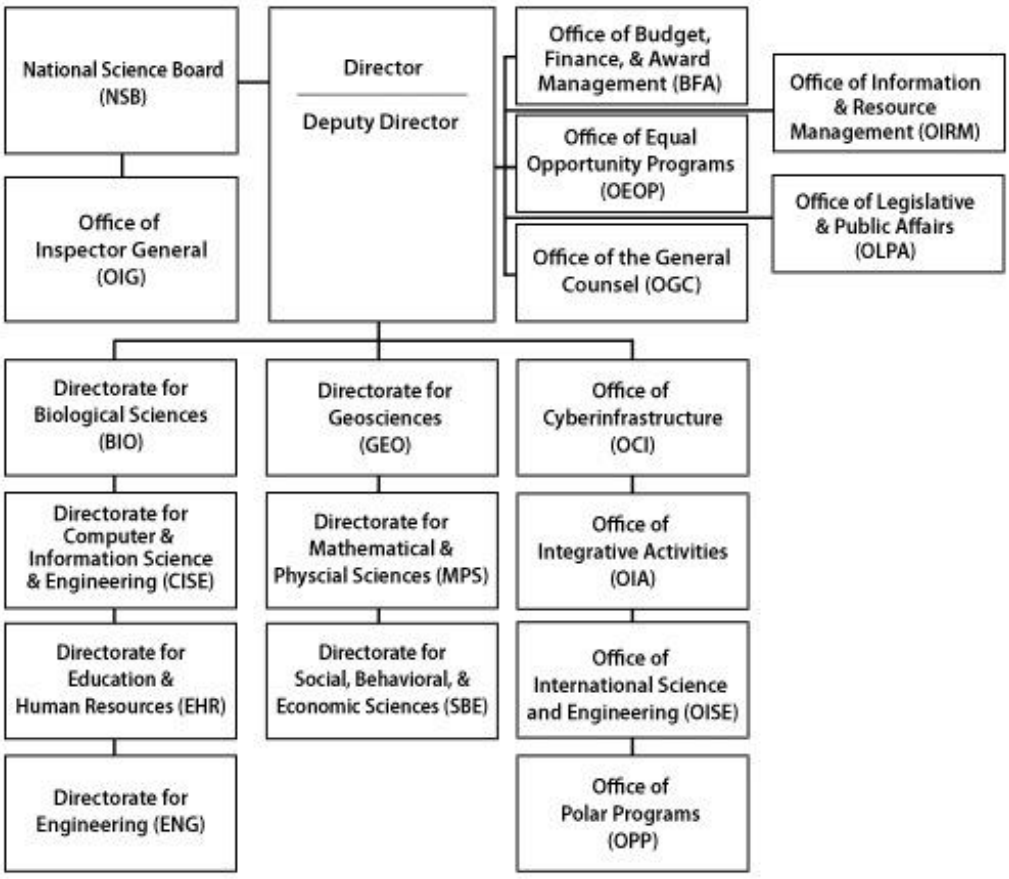

Figure A.1: NSF Organization Chart (Organization Chart, 2010)

The NSF (2010) funds research and education in almost every field of science and engineering. Faculty members interested in obtaining NSF funding for a research or a project submit their proposals to the NSF. To ensure that proposals are fairly evaluated, each proposal is reviewed by a NSF program officer and a panel of experts who are

independent of the NSF. After analyzing the potential of each project, the program officer makes an "award" or "decline" recommendation to the division director. The final decision on a proposal is generally completed at division level (How We Work). Figure A.2 below shows the flow of proposals during the approval process in detail.

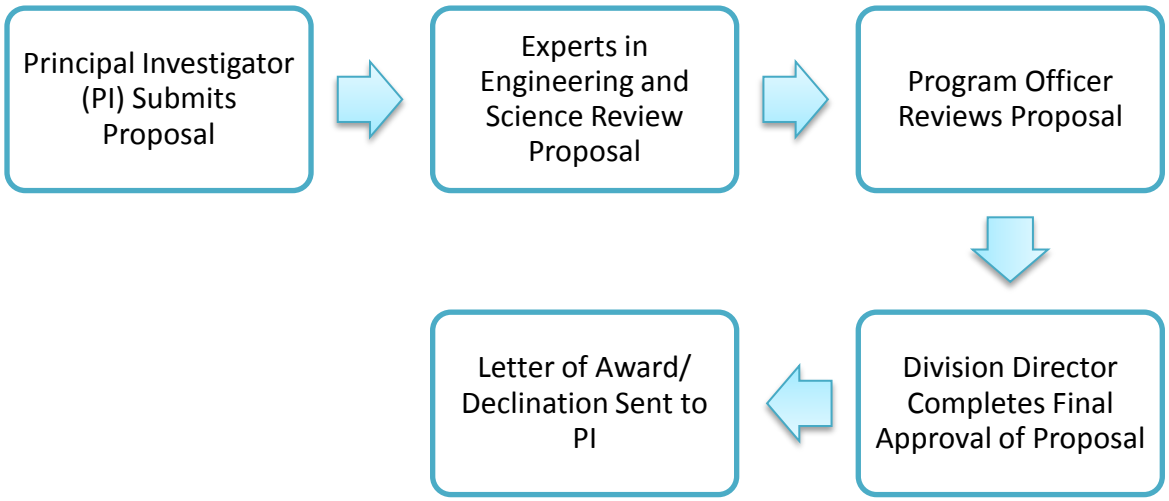

Figure A.2: NSF Flow of Proposals (Compiled by the author from "About NSF: How We Work", 2010)

There are approximately 40,000 proposals received each year for research, education and training projects, and approximately 11,000 of the total are funded (About Funding, 2010). "The high-risk, potentially transformative investments we make generate important discoveries and new technology, create and train a dynamic workforce, and spark the curiosity and creativity of millions" (FY 2010). Comparison of the total NSF budget appropriations in recent years is shown in Table A.1, and the trends of the appropriations in these years follow in Figure A.3.

Table A.1: NSF Budget Appropriations Comparison 2000-2009
(Compiled by the authors from "About NSF: Budget: NSF Budget Requests to Congress and Annual Appropriations", 2010)

Year	Total Appropriations ($ in Millions)	Increase in Appropriations (%)	Increase in Appropriations ($ in Millions)
2000	3912.00	+ 7.00	+ 24.000
2001	4416.38	+ 12.9	+ 504.38
2002	4789.30	+ 8.44	+ 372.92
2003	5309.95	+ 10.9	+ 520.65
2004	5577.85	+ 5.05	+ 267.90
2005	5472.82	- 1.88	- 105.03
2006	5581.17	+ 1.98	+ 108.35
2007	5917.16	+ 6.02	+ 335.99
2008	6065.00	+ 2.50	+ 147.84
2009	6490.40	+ 7.01	+ 425.40

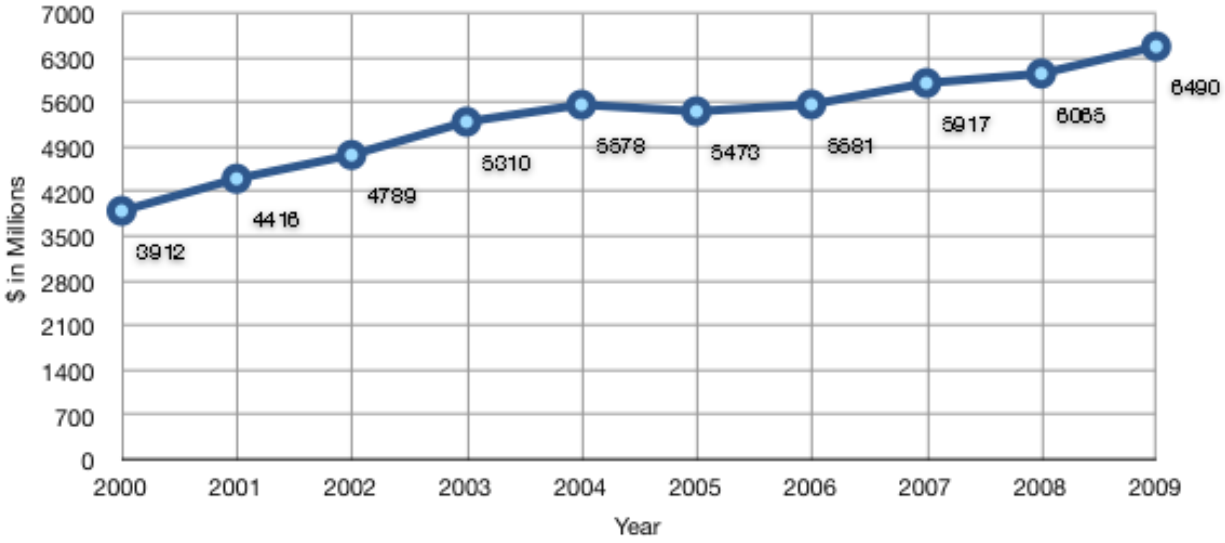

Figure A.3: NSF Budget Appropriations Trends 2000-2009 (Compiled by the authors from "About NSF: Budget: NSF Budget Requests to Congress and Annual Appropriations", 2010)

The Division of Engineering Education and Centers (EEC) of the Directorate for Engineering encourages partnerships between research and education. The major goal of the EEC division is to promote the successful education of engineers who can adapt well to the fast evolving technical environment in today's society (ENG-EEC-About EEC, 2010).

The EEC division measures the quality of the NSF's investments to assure the efficiency of public funds. They funded the Institute for Defense Analyses (IDA) Science and Technology Policy Institute (STPI) to facilitate a workshop for six months on evaluation in 2008, and then to provide the NSF's Engineering Education program with a review of outcomes of active grants in 2009 (EEC Program Evaluations, 2010).

In addition, to ensure successful engineering education, the following NSF (2010) programs have also recently evaluated students, teachers, and undergraduates:

- Faculty Early Career Development Program (CAREER)

- Bioengineering and Bioinformatics Summer Institute (BBSI)
- Nanotechnology Undergraduate Education (NUE)
- International Research and Education in Engineering (IREE)
- Innovations in Engineering Education, Curriculum, and Infrastructure (IEECI)
- Research Experiences for Undergraduates (REU) Program
- Research Experiences for Teachers (RET) in Engineering Program

Appendix B: List of Acronyms

ABET Accreditation Board for Engineering and Technology

AWCR Age Weighted Citation Rate

Co-PI Co-Principal Investigator

CV Curriculum Vitae

EEC Engineering Education and Centers (Division of the NSF)

GPRA Government Performance and Results Act

GUI Graphical User Interface

HTML HyperText Markup Language

IDA Institute for Defense Analyses

NSB National Science Board

NSF National Science Foundation

PDF Portable Document Format

PI Principal Investigator

STPI Science and Technology Policy Institute

TXT (plain text document)

URL Uniform Resource Locator

US United States

Appendix C: Glossary

Actor	See “node”
Beta	In terms of software development, a software tool that is still in the process of having bugs removed
Betweenness Centrality	The measure of which ties are most central within a social network
Closeness Centrality	A measure of how close an actor is to other actors within a social network
Crawl	The process a web crawler uses to compile data; the web crawler searches web pages for specific information and downloads pages containing that information to an offline database on the user’s computer
Data Pull	In reference to web crawlers, the action of collecting data from the pages being crawled
Database Management System	(DBMS) A system that controls and maintains a database, and provides an interface between a database and a user
Database	A digital collection of data
Degree	A measure of how many ties a node has within a social network
Filter Ability	In terms of a web crawler, the relative ease with which the user can filter the results of a crawl
Formative Evaluation	Evaluation that is used in the early stages of a project to evaluate initial and ongoing project activities
Freeware	Free software
HTML	(HyperText Markup Language) The predominant language for web pages
Interaction	Any sort of communication among individuals, groups of individuals, organizations, etc.
Logic Model	A conceptual model which describes individual elements of the project, the connections among them, and the expected outcomes
Node	A fundamental unit within a social network analysis visualization that represents an individual within the network

Page Rank	A type of rank that is specific to web crawlers (such as Google.com) based on the number of hits a webpage has
PDF	(Portable Document Format) A format in which users can save documents
Plain Text	A format in which users can save documents with no complex formatting; the best format to have a document in for text analysis tools
Politeness Protocol	Contains guidelines that minimize the harmful effects, such as crashing a web page, of a web crawler on the World Wide Web
Positive Impact	Some sort of change that meets the NSF's goals of funding the given project
Qualitative Analysis	Analysis that is based on data that are difficult to measure, such as opinions or actions
Qualitative Data	Data that are recorded or measured to describe the quality of something in size, appearance, or value
Quantitative Analysis	Analysis that is based on definite, easily measurable data, such as numbers
Quantitative Data	Data that are easily measurable, such as a number
Rank	How close to the beginning of a search a result of that search was (first, second, etc.)
Relation	See "Ties"
Revisit Protocol	Explains how to obtain the most up to date information from the World Wide Web
Search Depth	The number of pages away from the root URL that will be crawled by a web crawler
Seed Page	The initial list of URLs to visit that a programmer provides to a web crawler
Selection Protocol	Tells a web crawler which web pages to search and the order of searching
Social Network Analysis	A powerful data analysis tool that involves linking individuals through their interdependencies on other individuals
Square Data Array	See "Square Matrix"

Square Matrix	A matrix that contains the same number of rows as it does columns. In this report, this is the form in which data imported into a social network analysis software tool is organized before analysis
"Star" Network	One of the most common computer network topologies and the main component of more complex social networks
Summative Evaluation	Evaluation that is used in the later stages to assess the overall impact and quality of a completed project
Tag Cloud	See "Word Clouds"
Text Comparison Software	Software that compares nearly identical plain text documents and clearly marks where the differences between those documents are located
Ties	The connection between two nodes in a social network analysis visualization
TXT	See "Plain Text"
URL	(Uniform Resource Locator) The address of a page on the Internet
Web Crawler	A program that takes advantage of the branch structure of the World Wide Web by moving from page to page while collecting information
Word Cloud	A visual that emphasizes the relative frequency of a word within a sample of text by its size
World Wide Web	The Internet

Appendix D: Word Scrambling Process

We considered the possibility that the NSF may at some point wish to use word clouds to analyze documents that are not publically available. In response to this realization, we developed a procedure for making the document unreadable while keeping the words within the document intact. Below is the procedure we developed.

First, to reduce the number of columns in the Microsoft Excel spreadsheet (by reducing the number of words in each row):

1. Save the document in Notepad
2. Change the setting in Notepad to wrap the text in the window (Format -> Word Wrap)
3. Insert a carriage return (Enter) at the beginning of each line

Then, to use Microsoft Excel (2007) to scramble the words:

1. Import the text into Microsoft Excel
    a. Select the "Data" tab at the top
    b. Click "From Text" (on the left side of the Ribbon, in the section titled "Get External Data")
    c. Find the text file and click "Import"
    d. Under "Original Data Type", select the radio button for "Delimited"
    e. Click "Next"
    f. Select delimiters
        i. "Space" is an important one to use because it will guarantee that each word is in a different cell

ii. Other delimiters can be used in addition to “Space” if desired

g. Click “Next”

h. Under “Column Data Format”, select the radio button for “Text”

i. Click “Finish”

j. Click “Ok”

2. Alphabetize each column individually

a. Select the “Data” tab at the top, if not already selected

b. Click on the letter at the top of a column containing words (Ex: Column “A”)

c. Click the button in the section titled “Sort & Filter” that has an “A” on top of a “Z” beside an arrow pointing down. This button will sort the row by letter from A to Z.

i. If a box that says “Microsoft Excel found data next to your selection. Since you have not selected this data, it will not be sorted.” appears and gives you the choice to either “Expand the selection” or “Continue with the current selection”:

1. Select “Continue with the current selection”

2. Click “Sort”

d. Repeat a-c until all columns have been sorted

3. The document should now be unreadable

4. Select all the cells that contain words

5. Copy the words

6. Paste the words into an online word cloud generator (we recommend Tagxedo)

7. Select option within the word cloud generator to create visualization

a. This will vary depending on the word cloud generator

Appendix E: List of Evaluated Software Tools

**Best of Category*

Data Compilation

Web Crawlers

Website Ripper Copier *

http://www.tensons.com/products/websiterippercopier/

Webreaper

http://www.webreaper.net/

Visual WebRipper

http://www.visualwebripper.com/

Databases

Publically available NSF Database

http://www.nsf.gov/awardsearch/

Authorship Network Data: ISI Web of Knowledge

http://apps.isiknowledge.com/UA_GeneralSearch_input.do?product=UA&search_mode=GeneralSearch&SID=3BMmOahlB7lm8JccJk2&preferencesSaved=

Data Organization

Text Conversion Tools

Some PDF to Word Converter (PDF to Plain Text)*

http://download.cnet.com/Some-PDF-to-Word-Converter/3000-2079_4-10836743.html

Zilla PDF to TXT Converter (PDF to Plain Text)

http://www.pdfzilla.com/zilla_pdf_to_txt_converter.html

Text Mining Tool (HTML to Plain Text) *

http://www.softpedia.com/get/Office-tools/Other-Office-Tools/Text-Mining-Tool.shtml

Detagger (HTML to Plain Text)

http://www.jafsoft.com/detagger/

Data Analysis and Visualization

Text Comparison Tools

KDiff3 *

http://kdiff3.sourceforge.net/

ExamDiff

http://www.prestosoft.com/edp_examdiff.asp

WinMerge

http://winmerge.org/

Word Cloud Generators

Tagxedo *

http://www.tagxedo.com/

Wordle

http://www.wordle.net/

TagCrowd

http://tagcrowd.com/

TagCloudMaker

http://webscripts.softpedia.com/script/Snippets/TagCloud-Maker-10219.html

Social Network Analysis Tools

UCINET

http://www.analytictech.com/ucinet/

Gephi *

http://gephi.org/