Dynamic Network Analysis

Kathleen M. Carley and CASOS Students and Staff
This is a preview copy.

Many people are involved in the development of this book – Among the authors are – Kathleen M. Carley, Matt deReno, Jamie Olson, Terrill Frantz, Jana Diesner, Brian Hirshman, George Davis, Peter Landwehr. This list is not complete. All changes, suggestions, corrections should be sent to kathleen.carley@CS.CMU.EDU WITH THE HEADER dna book changes

CHAPTER 1: The Essence of Network Analysis

We have come a long way since the days of plotting network connections on bulletin boards with push pins. Our models today are visually more engaging but more so than a simple visceral improvement, we have come to think of networks in a wholly new scientific manner. The power of mathematics, statistics, and computer science has now been applied to network analysis.

Traditional approaches to network analysis focused on the social network – who interacts with whom. Indeed most of the traditional measures and our understanding of interpreting them are based on decades of research looking at such networks. Today, however, we find the need to move beyond such networks to ask, what is the context in which they operate? How does who you know impact what you know? What you do? Where you do it? In other words, networks of interaction are now embedded in complex meta-networks that link who, what, how and why through time and space. Dynamic Network Analysis the study of such networks. In this book we will move from the basics of social network analysis to the more detailed dynamic network analysis.

Areas where dynamic network analysis can be applied are varied and nearly limitless. Gaining an understanding of the structure of Al Qaeda is critical in fighting the war on terror and could help prevent future events such as another 9-11; Possessing a true ecological map of a food chain will help keep environments stable (Johnson 2001); Because of limited resources, understanding the varied shipping lanes merchant marine vessels traverse as they conduct international trade is vital to protecting ports of call (Davis and Carley); Understanding how a network of satellites is connected to various locations around the world is critical for a global company's bottom line; A financial network, such as those that enabled the fraud at Enron to destroy the entire company and make a lifetime's retirement fund disappear in a day, are all fertile territory for dynamic network analysis.

Networks are all around us. Network models attempt to paint a picture of reality that is otherwise too complex for the human mind to comprehend. Networks pervade everything and in that sense, networks are inescapable. You cannot go through life without belonging to a network of some sort, even different vastly networks, multitudes of networks, which are all interconnected to other sorts of networks of which you may or may not have a clue as to their very existence. Even if you are not associated with one particular network, you can still be said to be isolated from the network, therefore part of the network by virtue of subtraction. Therefore, networks are part of the firmament.

This begs the question then what is the best way to analyze a network as complex as a food chain, a nefarious operation to blow up a U.S. Embassy in Tanzania, to murder Julius Caesar? The answer lies in the science of Dynamic Network Analysis, the most robust rich approach to network analysis today.

Why is Dynamic Network Analysis the answer?

DNA (Dynamic Network Analysis) is concerned with not only Who is connected to Who as in traditional Link Analysis (LA), but also How they are connected is a major factor in DNA. Moreover, there are a multitude of questions DNA attempts to answer: Where they are connected, What is exactly connected, How their connections change over time, How they are likely to evolve, When and Where connections exists; In short, Why Who is Where When and Who else is with Who and What do they all share in common and How do they plan to accomplish their goals – are you getting bewildered and confused yet? Indeed, the possibilities can be mind-boggling because networks can be mind-boggling because of their complexity (Carley 2004). DNA simplifies networks. It cuts through and clarifies what is otherwise a jumble of interconnected points of interest.

Our plan then in this book is to introduce the exciting field of Dynamic Network Analysis to the uninitiated as well as to the initiated. It is our broad and lofty hope that you will soon find yourself immersed in DNA, thus able to reap the powerful, insightful and critical analysis that only DNA makes possible. DNA is a science whose practicality and application is boundless. Nearly anything under the

sun can be put under the microscope of Dynamic Network Analysis. New, more insightful, innovative revelations can be presented heretofore impossible under the traditional methods of Link Analysis. That is the power of DNA.

In this book, we will explore that power and in doing so, approach William Shakespeare's Tragedy of Julius Caesar with the analytic precision of DNA. We will study this complex multifaceted plot with DNA to illustrate the power of DNA.

Who can use DNA

Dynamic Network Analysis can be used by any type of analyst interested in state-of-the-art network analysis for a variety of reasons. Such an analyst might work for the CIA, FBI or the IRS, large scale health system, a global food back, a computer network administrator or be a military analyst. The ubiquity of networks means nearly any organizational analyst can come to the table of DNA with the goal of solving his or her own unique network problem (Carley 2006, forthcoming) no matter what field the analyst is involved in and have a tool at his or her disposal to accomplish exactly that sort of task. But, due to its power, there are more apt network for DNA application. To understand what networks are best for DNA, we need to learn how DNA is applied and what process is involved in conducting a DNA analysis.

There are two aspects of Dynamic Network Analysis. The first is the statistical analysis of DNA data. The second is the use of simulation to address issues of network dynamics, which is learning about how networks evolve over time.

DNA networks vary from traditional social networks in that they are larger dynamic multi-mode, multi-plex networks, and may contain varying levels of uncertainty; uncertainty, just like that faced by Julius Caesar when presented the warnings of the Soothsayer.

Moreover, DNA statistical tools are generally optimized for large-scale networks and admit the analysis of multiple networks simultaneously in which, there are multiple types of entities (multi-entities) and multiple types of links (multiplex) ; In contrast, SNA statistical tools focus on single or at most two mode data and facilitate the analysis of only one type of link at a time (Freeman 2000). DNA is a far more robust solution to analyzing complex networks.

DNA statistical tools tend to provide more measures to the user, because they have measures that use data drawn from multiple networks simultaneously (Carley 2003; Carley 2003). From a computer simulation perspective, entities in DNA are like atoms in quantum theory, entities can be treated as probabilistic. Whereas entities in a traditional SNA model are static, entities in a DNA model have the ability to learn. Properties change over time; entities can adapt (Breiger 2003): A company's employees can learn new skills and increase their value to the network. Julius Caesar's "friends" "Romans" and "Countrymen" might learn new knowledge, change political allegiance and maybe, just maybe, stab him in the back when he is invited to the Senate floor – all over the course of time (Watts and Strogatz 1998).

DNA allows us to analyze the interplay between various different types of Who, What, Where, When, How, which are the entities that can pretty much include just about anything in the physical universe (Carley 1998). That is what DNA places at our fingertips. DNA adds the critical element of a network's evolution and considers the circumstances under which change is likely to occur and how it applies to Entities (Carley 2004). To begin, we need to know more about Entities.

Entities

An entity in DNA is essentially the *Who*, *What*, *Where*, *How* and *Why* – *something* that is being studied. In addition, as we mentioned, Networks can be made up of practically anything: you and your friends constitute a social network and in it as such represent Entities in that network. An ecosystem

comprised of millions of living organisms and innumerable food sources would be another example of network Entities. A company might set up a network of computers and thus the computers are now Entities in that network as well.

An isolated group of terrorist might constitute a cellular network, as each terrorist is an Entity of some sort. Orbiting satellites can be networked. Our solar system is a vast network of billions of stars of varying size and shapes; taken overall they comprise the Milky Way galaxy. The Milky Way galaxy is one of many galaxies comprising the infinitely vast network of galaxies that constitute the Universe. Are you beginning to notice a pattern? Have we made it clear that just about anything you think of can be considered a network on some abstract level and as such anything that make up a Network is an Entity of some kind?

We will get into more details about the sorts of Entities a DNA scientist typically deals with soon enough. For now, we want to consider another element that acts on everything with is just about an integral to an entity as the physical substance of which it is comprised. This element is Time. Time does not stand still and over any given period, things, that are Entities, are likely to change. Much like Albert Einstein made the connection between space and time. It is clear that networks occupy space and it would only make sense to see that integral to a networks space is the time in which it exists. Because, a network that exists this year but be dramatically different then the same network represented next year. DNA takes this change into account.

Entities are always changing, which makes entities dynamic. DNA is making inroads into studying structure heretofore inaccessible in the traditional disciplines of link analysis where change was not likely to be a key factor. DNA looks at networks not merely as a bunch of people connected to other people, but people that exist in time and can be different, and often are, from one time period to the next. Some people, after all, learn new knowledge and forgot knowledge just the same. We are looking at networks in terms of their interconnectedness to other entity networks and how change occurs as time marches on (Carley 2001) and this is where DNA proves its mettle. Because with DNA, we might take a snapshot of a network in time and with some skill in analysis, stay one step ahead of the curve.

Let us now consider DNA on a more practical level, in a manner that might help explain situations you have probably encountered many times. The importance of networking is something we have all encountered before in one form or another be it in our personal or professional lives. In such everyday experiences, we might say that "networking" is the art of making meaningful connections. We have all heard the expression: *It is not what you know but who you know.* Let us consider this common morsel of wisdom from perspective of the DNA.

More than "who you know"

Countless variations of the phrase "It is not what you know but who you know" ring true across many boundaries from the cynically hardened skeptics to the most incorrigible optimists. Moreover, such a turn of phrase is often ascribed as the key to both personal and professional success. However, what is this phrase really hinting at? We know it describes a network but what exactly about the network? The laws of sociological nature? The laws of social dynamics? How to get ahead? The art of networking? A certain part, or aspect, of what it means to be important component of a network? All of the above? None of the above? Some of the above?

The truth is that simple phrase, *It is not what you know, but who you know,* describes merely one single facet of any social network and the DNA analyst would find this turn of phrase incredibly simplistic. In fact, have you ever considered it might be totally wrong? Can you think of examples where the opposite might be true? How about a network model of a PhD program where an advanced degree is conferred by accumulating and presenting research and defending such research until the thesis is accepted? In this network, could it not be argued that *what you know* is more important than *whom you know*? Perhaps. Nevertheless, back to that hackneyed phrase: *It is not what you know, but who you know*. What is to be really made of it from the stand point of DNA?

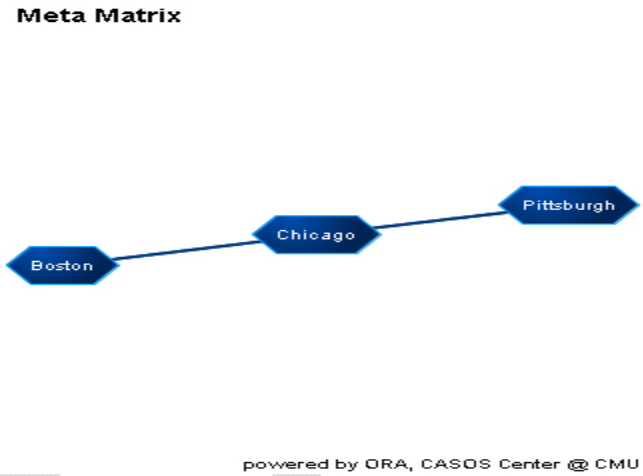


Figure 1: Airports Hubs connected

It some ways that tired phrase, "It is not what you know, but who you know," is tantamount to something as analytically awkward as, "*It is the steering wheel that enables a car get to its destination*." Let us consider this analogy briefly.

We can all comprehend that on some level a steering wheel, or other sort of mechanism, is needed to get a car to its destination. The problem then with the statement "It is the steering wheel that enables a car to get its destination" is that this statement makes no consideration about the thousands of other aspects of a car that make the vehicle function: the bolts, nuts, pistons, hoses, and springs; seats, breaks, transmission parts; the axels, frame and chassis; not to mention you need a suitable road on which to drive the vehicle to get anywhere at all? You also need to be licensed to drive which means a driver must have a certain amount of minimal knowledge to operate the automobile. So, what does it say then about the statement: "*It is the steering wheel that gets a car to its destination?"* It sounds silly because it is obvious so much is missing.

On a similar token, summarily stating success comes down to the statement: "*It is not what you know, but who you know"* you may as well be stating "*It is the steering wheel that gets the car to its destination*." To a DNA analyst, this statement is silly because again, it is obvious so much is missing.

However, one might argue that there is a certain practicality conveyed in the original statement about whom you know being somewhat important to level of your desired success. Moreover, the statement seems to abstractly describe *something* useful about social networks on an intuitive common sense level. It could be argued the statement resonates at such a level because much of our understanding of networks is intuitive to some degree. Perhaps it might not be wise to discount the statement completely.

That may be true. But what is true as well is that the science of DNA would look at that statement as merely a small tip of a network iceberg heretofore impossible to glimpse until DNA arrived on the scene. Ignore icebergs at your owner peril. They have a tendency to sink ships and your notion of a network just might be one of them. The statement *"It is who you know, not what you know"* is only once small facet of a network model approached with DNA. Now let us consider an example of just how limiting old network thinking is. We will continue onward with the statement "It is not what you know but who you know…" It seems to serve a good purpose here.

A traditionally limited view of networks

Let us humor a decision making model based on evaluations within the context of the statement *"It is not what you know but who you know"*. In this example, we will demonstrate that believing such statements to be true can easily result in erroneous conclusions based on such simple network models and philosophies for that matter.

For instance, assume that most stay at home mothers have no interest in and thereby have a limited knowledge of how to invest in the stock market (a gross stereotype). We will say then you have a stereotype about stay-at-home mothers that clouds your decision making when it involves anyone fitting this preconception of what a stay-at-home mother does. Now let us say you are in need of good financial advice involving a recent investment decision you need to make. So what do you do?

In your narrow mind which looks for narrow network models to analyze, you make a choice that you will seek some financial help from your good friend Sam because Sam is a financial advisor and well respected. You stop over at Sam's house to see if he is home. You goal is for Sam to give you his opinion on what would be the wisest decision to make regarding your investment decision. You know Sam to be very knowledgeable in the areas of investing and you trust his judgment as sound. Unfortunately, because you failed to consider time as an element of your network, it looks like you missed him. Sam is out. So now what do you do?

Instead, his wife, Sarah, a stay- at-home mother, answers the door and tells you he is away on vacation and can't be reached. Now, you have a resource tie is that is severed. Because of you proclivity to think along simple network statements, you sigh learning Sam will not be available and head back home relegating yourself to the best guess you can make about your investment. However, did you miss something here?

Little did you know that Sarah, a former investment banker, is simply taking a few years off to raise her children? That is she has the "knowledge" that you were looking for in Sam. Even then, she works as an investment consultant from home and even lends Sam advice when it comes to his own client's portfolio. She has the knowledge you didn't know she had. Sam on the other hand, has a position of financial importance but is not the primary resource behind all his wise solutions – his wife is. How did you miss it? Because you adhered to a limiting model of networking about stay at home mothers. You did not have an accurate picture of the network model nor did you possess an accurate picture about how "What" they know connects to "Who" you know. Your network statement failed you. The DNA analyst would likely have not made the same mistake because from the get go, the manner in which DNA would have been applied to evaluating this network.

Once again, we go back to our phrase the quintessential stereotypical opinion of networking: *It is not what you know, but whom you know.* Does that phrase sound very practical anymore?

You just moved to a new town and you only know a handful of people? You need to find a babysitter on a quick notice, so you and your spouse can have a night out and enjoy quiet time together. What is your decision process? Intuitively, you will construct a mental map – a network so to speak – of the steps you need to complete to find a babysitter. You will need to tap into a social network to accomplish this task. Most likely, you will look for a phone book or perhaps perform a Google search on "babysitters" or start asking around. Without even thinking about it, you have given yourself the goal of finding and locating a babysitter and you making a network evaluation. Perhaps you will call somebody you know, like Sheila, one of your close friends. You might be inclined to search locally in town or get a recommendation from Sheila, who might be able to make a wise suggestion.

Let us say Sheila does not know any babysitters. This friend, for the sake of your network, doesn't know anybody at all, insofar as it relates to babysitting. However, your other friend, Bob, *does* know somebody *that knows* a babysitter (i.e., the friend of a friend). Now we have it that there are two people

connected (links) between you and this babysitter. We would say, at this point at least, your network is comprised of *You, Bob, Bob's friend, and The Babysitter*. In fact, we just described something a network analyst would call a "path" from you to your babysitter. You call the babysitter and are impressed with their experience. You hire the babysitter. Your goal has been completed. To accomplish this task, you plugged into a social network and made a decision.

Was the decision process intuitive and logical? For the most part, yes it was. However, without even realizing it, you evaluated several key components of the social network which sociologists and network analysts have deemed useful in network analysis science: They are "agents," in this case your friends; "resources," such as the Google search engine and the phone book; "knowledge" in that somebody needed to know where to find a babysitter; and "tasks," which basically is your goal in finding one. Then there is the "location" of your babysitter. After all, we have to know where to find the babysitter geographically. You also took into consideration the time the babysitter would be available. After all, a sitter with the knowledge and resources is useless to you if they happen to be on vacation next month when you need them.

Let us imagine you are the president of your own company of 45 professionals who work in the areas of software development to marketing to research and distribution. Things were going well for years, but now profits are slipping. You begin to wonder if you are maximizing the "resources" you have at your disposal. Do you truly have the most talented people in the most important roles critical to your business? Is it time to look at how the skills in your company have changed? What if years go by and everybody has the same knowledge they did when they started. Chances are your knowledge base would be become outdated and obsolete. You need to make sure that does not happen so you plan to periodically monitor your employee's talent and knowledge base.

You need a tool to figure out if your company is set up in the best way possible but are not sure how to do it. So what do you do? Do you just take an educated guess about how your company should be arranged to maximize employee's talents, skills and resources? Wouldn't it be nice if there were a tool that could make a model of your company and based on the linkages of education, aspirations, responsibilities, access to resources, business skills, to tell what your company what it really looks like? For instance, wouldn't you like to know who would constitute the weakest link in your company or the most underused employee from a standpoint of their knowledge? Who are the emerging leaders? What employees have the most knowledge and where are they located? Can they access the right resources for the tasks they are given?

Perhaps you want to study how your company might likely perform based on the removal of "John" since John is moving to an out of town job. Who might take his place? Does he have access to some resources that nobody else has access too? Was he a "silo" of special information? What if John performed critical functions that officially belonged to someone else? How would you know that? Along those lines, what are the informal channels that exist at your company? Does your catering manager actually know the whereabouts of your key personnel better than then the respective executive assistant does?

We can't answer any of the aforementioned questions by simply making educated guesses about who really does what in a haphazard fashion. We need a scientific method to go about a true analysis of the company. We need to create a model that takes into account all the critical entities and how they are truly connected. We would like our tool to provide scenario-based analysis as well. Moreover, we need it to consider incomplete information or information that might be outdated. Could such a tool allow us to even predict how your company might grow in the near future given the removal or addition of any key node? The answer is DNA.

Everyone that has ever heard of a terror network or complex organizational structure has an intuitive idea about how such a network might be displayed and hence analyzed. Such a person might logically

envision that any such terror network might have a leader and a group of underlings to carry out certain terror-oriented tasks. They may further conclude such a cell could be plotted out on paper by denoting actors with dots and drawing lines between them symbolizing connections. Likewise, a complex organization might have a hierarchical structure with a president and board of directors sitting at the top. However, very few people, who are not trained in the science of Dynamic Network Analysis, will have an inkling of an idea of what can be fully gleaned from network analysis when one takes into account the cross-disciplinary approach of computational mathematics and other social-science disciplines.

In such a science, factors that are more complex are considered when conducting network analysis. For instance, much like the *Special Theory of Relativity* changed the way we think of *space* and *time* to something called *space-time*, we have to take a far deeper analytical approach to what we mean when we say network analysis. After all, networks are not like molecules – they can learn. Yes, that is right. We already went into how Networks can be comprised of nearly anything. One thing you are well familiar with is that Networks, the ones we are most often interested in analyzing, are made up of people and those people have a tendency to learn and forgot, grow and decline. People also tend to react to certain events in different ways, which could easily change a network model.

So let us be clear on this point: networks don't exist in a vacuum – they evolve (Bonacich 2001). Networks don't suffer damage without responding in some way – be it growth or the emergence of new leaders and increased activity. Networks don't stay the same either – they change constantly. What you analyzed today is altered by the time it is read and considered by another. The change can be dramatic or it can be small but any change can be critical. One's assessment of any part of the network can be skewed if the information, on which your assumptions are based, proves to be false. Along similar lines, the information you have on a network might only be the tip of the iceberg. When you consider all these quandaries, a more robust science is needed to carry out effective network analysis.

Below is a list of issues that can be tackled with Dynamic Network Analysis:

- Developing metrics and statistics to assess and identify change within and across networks.
- Developing and validating simulations to study network change, evolution, adaptation, decay...
- Developing and validating formal models of network generation and evolution
- Developing and testing theory of network change, evolution, adaptation, decay...
- Developing techniques to visualize network change overall or at the node, the representation of a single entity, or group level, which contains multiple entities.
- Developing statistical techniques to see whether differences observed over time in networks are due to simply different samples from a distribution of links and nodes or changes over time in the underlying distribution of links and nodes
- Developing control processes for networks over time.
- Developing algorithms to change distributions of links in networks over time.
- Developing algorithms to track groups in networks over time.
- Developing tools to extract or locate networks from various data sources such as texts.
- Developing statistically valid measurements on networks over time.
- Examining the robustness of network metrics under various types of missing data.
- Empirical studies of multi-mode multi-link multi-time period networks.

- Examining networks as probabilistic time-variant phenomena.
- Forecasting change in existing networks.
- Identifying trails through time given a sequence of networks.
- Identifying changes in node criticality given a sequence of networks and anything else related to multi-mode multi-link multi-time period networks

(Aldrich and Herker 1977; Wasserman 1980; Watts 1999; Carley 2001; Carley 2003; Carley 2003; Carley 2003; Carley 2004; Carley 2004).

The Tragedy of Julius Caesar

To understand DNA more fully, we will apply the tool to the Tragedy of Julius Caesar as crafted by William Shakespeare. Why Julius Caesar? In short, we have chosen the Tragedy of Julius Caesar because chances are it is a literary work many of us have encountered at one time or another in our educational backgrounds whether in High School or at the Post-secondary level. Moreover, it is about a simple usurpation of power, an assassination, a betrayal and conflicting values, a plot, a network of people who made decisions. It is about love of country and love of your fellow man. There is a lot of network complexity in that play. Therefore, in our opinion, The Tragedy of Julius Caesar is a highly useful network from that standpoint: neither too big nor small. It is just right to show the power of DNA properly with a model that you may likely know already, one made up of a fairly complex arrangement of characters, allegiances and resources. Might we even suggest to Julius Caesar how he could have prevented his own demise? If we had a time machine, we just might do that with DNA.

You need not re-read the play to understand the examples we will go through in our application of DNA. However, a familiarity with The Tragedy of Julius Caesar, might help you get more out of this book. We suggest *Sparknotes.com* for short but concise summary of the characters, events and plot (simply search "Julius Caesar" on the SparkNote's site search function). You can also get a copy of *Cliff's Notes* from your local bookstore. Better yet, purchase a copy of the play, dust off the old one in your book collection, and do something novel, like read the play over again. It shouldn't take you more than a couple hours. You might even enjoy it.

It is our hope that presented with the proper DNA model; even Brutus would have seen that the fault surely did not "lie in the stars" as Cassius reminded him in course of events. Rather, the fault lies in the failure to analyze a complex multimodal network of Roman politicians, plebeians, military leaders, poets, family member, citizens, soothsayers, battles, skills, allegiances, knowledge, rhetoric and what have you—this is where true fault resides.

So we begin a journey, in hindsight nonetheless, to analyze The Tragedy of Julius Caesar by William Shakespeare, and offer our own analytical recommendations and insights surrounding Caesar's assassination by putting the power of Dynamic Network Analysis to work on the network of Julius Caesar as extrapolated from the legendary play The Tragedy of Julius Caesar.

Plot Overview: The Tragedy of Julius Caesar

Let us begin to consider DNA in the context of our specially created example solely for the illustrative purposes of this book: *The Tragedy of Julius Caesar by William Shakespeare*.

The Tragedy of Julius Caesar by William Shakespeare portrays the conspiracy against the Roman dictator Julius Caesar, his assassination and its aftermath. It is one of several Roman plays that he wrote, based on true events from Roman history, which also include Coriolanus, Antony, and Cleopatra among others. Although the title of the play is Julius Caesar, he is not the central character in its action as you

will soon learn. In fact, Julius Caesar appears in only three scenes and dies at the beginning of the third act. The protagonist of the play is really Marcus Brutus and the central psychological drama is his struggle between the conflicting demands of honor, patriotism, and friendship.

If you must know, it is believed the play reflected the general anxiety of England due to worries over succession of leadership. At the time of its creation and first performance, Queen Elizabeth, a strong ruler, was elderly and had refused to name a successor, leading to worries that a civil war similar to that of Rome's might break out after her death. Well, you might as well take this play and apply it as a network model to any period to capture the angst of a generation undergoing dramatic political change.

As the play begins two tribunes, Flavius and Murellus, find masses of Roman citizens roving the streets, taking the day off in order to gawk and praise Julius Caesar's triumphal parade. And why shouldn't they: Caesar has defeated the Roman general Pompey, his archrival, in battle. The tribunes berate the citizenry for abandoning their duties and order them to remove decorations from Caesar's statues. Caesar enters with his entourage, including the military and political figures Brutus, Cassius, and Antony, all the major figures in the rest of the story. A Soothsayer calls out, quite famously, "beware the Ides of March," but Caesar ignores him as he will many other dire predictions and continues with his victory celebration.

Cassius and Brutus, both longtime confidantes of Caesar and each other, converse. Cassius tells Brutus that he has seemed aloof lately; Brutus replies that he has personal doubts and is very troubled. Cassius states that he wishes Brutus could see himself as others see him, for then Brutus would realize how honored and respected he is and know his rightful place. Brutus says that he fears that the people want Caesar to become king, which would overturn the republic and convert it over into a despotic regime. Cassius agrees that Caesar is already worshiped like a god. Cassius reminds Brutus that Caesar is merely a man, no better than Brutus or Cassius.

Cassius recalls incidents of Caesar's physical weakness and marvels that this fallible man has become so powerful. He blames his and Brutus's lack of will for allowing Caesar's rise to power. Brutus considers Cassius's expressions as Caesar returns. Upon seeing Cassius, Caesar tells Antony that he deeply distrusts Cassius.

Caesar departs, and another politician, Casca, tells Brutus and Cassius that, during the celebration, Antony offered the crown to Caesar three times and the people cheered, but Caesar refused it each time. He reports that Caesar then fell to the ground and had some kind of seizure before the crowd; his demonstration of weakness, however, did not alter the plebeians' devotion to him. Brutus goes home to consider Cassius's words regarding Caesar's poor qualifications to rule, while Cassius hatches a plot to draw Brutus into a conspiracy against Caesar.

That night, Rome is plagued with violent weather and a variety of bad omens and portents. Brutus finds letters in his house apparently written by Roman citizens worried that Caesar has become too powerful. The letters have in fact been forged and planted by Cassius, who knows that if Brutus believes it is the people's will, he will support a plot to remove Caesar from power. A committed supporter of the republic, Brutus fears the possibility of a dictator-led empire, worrying that the populace would lose its voice. Cassius arrives at Brutus's home with his conspirators, and Brutus, who has already been won over by the letters, takes control of the meeting. The men agree to lure Caesar from his house and kill him. Moreover, Cassius wants to kill Antony too, for Antony will surely try to hinder their plans, but Brutus disagrees, believing that too many deaths will render their plot too bloody and dishonor them. Having agreed to spare Antony, the conspirators depart. Portia, Brutus's wife, observes that Brutus appears preoccupied. She pleads with him to confide in her, but he rebuffs her .

Caesar prepares to go to the Senate. His wife, Calpurnia, begs him not to go, describing recent nightmares she has had in which a statue of Caesar streamed with blood and smiling men bathed their hands in the blood. Caesar refuses to yield to fear and insists on going about his daily business. Finally,

Calpurnia convinces him to stay home—if not out of caution, then as a favor to her. But Decius, one of the conspirators, then arrives and convinces Caesar that Calpurnia has misinterpreted her dreams and the recent omens. Caesar departs for the Senate in the company of the conspirators.

As Caesar proceeds through the streets toward the Senate, the Soothsayer again tries but fails to get his attention. The citizen Artemidorus hands him a letter warning him about the conspirators, but Caesar refuses to read it, saying that his closest personal concerns are his last priority. At the Senate, the conspirators speak to Caesar, bowing at his feet and encircling him. One by one, they stab him to death. When Caesar sees his dear friend Brutus among his murderers, he gives up his struggle and dies.

The murderers bath their hands and swords in Caesar's blood, thus bringing Calpurnia's prediction to fruition. Antony, having been led away on a false pretext, returns and pledges allegiance to Brutus. However, afterward, he weeps over Caesar's body. He shakes hands with the conspirators, thus marking them all as guilty while appearing to make a gesture of conciliation. When Antony asks why they killed Caesar, Brutus replies that he will explain their purpose in a funeral oration. Antony asks to be allowed to speak over the body as well; Brutus grants his permission, though Cassius remains suspicious of Antony. The conspirators depart, and Antony, alone now, swears that Caesar's death shall be avenged.

Brutus and Cassius go to the Forum to speak to the public. Cassius exits to address another part of the crowd. Brutus declares to the masses that though he loved Caesar, he loves Rome more, and Caesar's ambition posed a danger to Roman liberty. The speech placates the crowd. Antony appears with Caesar's body, and Brutus departs after turning the pulpit over to Antony. Repeatedly referring to Brutus as "an honorable man," Antony's speech becomes increasingly sarcastic; questioning the claims that Brutus made in his speech that Caesar acted only out of ambition, Antony points out that Caesar brought much wealth and glory to Rome, and three times turned down offers of the crown. Antony then produces Caesar's will but announces that he will not read it for it would upset the people inordinately. The crowd nevertheless begs him to read the will, so he descends from the pulpit to stand next to Caesar's body. He describes Caesar's horrible death and shows Caesar's wounded body to the crowd. He then reads Caesar's will, which bequeaths a sum of money to every citizen and orders that his private gardens be made public. The crowd becomes enraged that this generous man lies dead; calling Brutus and Cassius traitors, the masses set off to drive them from the city.

Meanwhile, Caesar's adopted son and appointed successor, Octavius, arrives in Rome and forms a three-person coalition with Antony and Lepidus. They prepare to fight Cassius and Brutus, who have been driven into exile and are raising armies outside the city. At the conspirators' camp, Brutus and Cassius have a heated argument regarding matters of money and honor, but they ultimately reconcile. Brutus reveals that he is sick with grief, for in his absence Portia has killed herself. The two continue to prepare for battle with Antony and Octavius. That night, the Ghost of Caesar appears to Brutus, announcing that Brutus will meet him again on the battlefield.

Octavius and Antony march their army toward Brutus and Cassius. Antony tells Octavius where to attack, but Octavius says that he will make his own orders; he is already asserting his authority as the heir of Caesar and the next ruler of Rome. The opposing generals meet on the battlefield and exchange insults before beginning combat.

Cassius witnesses his own men fleeing and hears that Brutus's men are not performing effectively. Cassius sends one of his men, Pindarus, to see how matters are progressing. From afar, Pindarus sees one of their leaders, Cassius's best friend, Titinius, being surrounded by cheering troops and concludes that he has been captured. Cassius despairs and orders Pindarus to kill him with his own sword. He dies proclaiming that Caesar is avenged. Titinius himself then arrives—the men encircling him were actually his comrades, cheering a victory he had earned. Titinius sees Cassius's corpse and, mourning the death of his friend, kills himself.

Brutus learns of the deaths of Cassius and Titinius with a heavy heart, and prepares to take on the Romans again. When his army loses, doom appears imminent. Brutus asks one of his men to hold his sword while he impales himself on it. Finally, Caesar can rest satisfied, he says as he dies. Octavius and Antony arrive. Antony speaks over Brutus's body, calling him the noblest Roman of all. While the other conspirators acted out of envy and ambition, he observes, Brutus genuinely believed that he acted for the benefit of Rome. Octavius orders that Brutus be buried in the honorable way. The men then depart to celebrate their victory.

Now that we got that out of the way, it is time to get down to some Dynamic Network Analysis. After all, we know the story, now we need to know the nodes, the Whos, that will make up our MetaNetwork. And, without further adieu, we are ready to talk about the basic building blocks of network analysis.

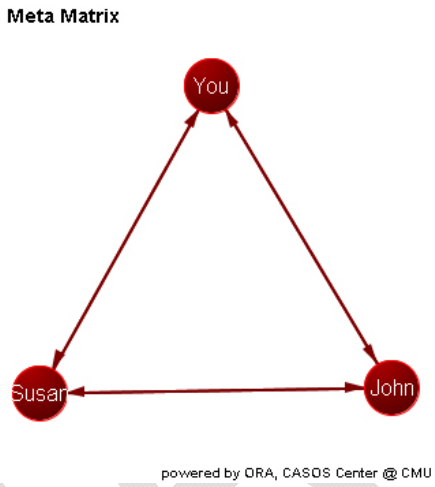


Figure 2: Small People Network of you, Susan and John: A traditional link analysis model. This model may be loosely accurate, but it is clear it only reflects such a relationship at one given point in time. Over time, relationships change and traditional link analysis models never accounted for this obvious fact. DNA ads the element of time.

What can be a network?

Nearly everything is a network. The universe is expanding. Your knowledge is growing or languishing. People move on to different roles. One day you're a son, the next day you are a parent. Like string theory and quantum mechanics, everything in our vast interconnected universe is, on some level, constantly on the move and this is what you will come to see in Julius Caesar. The time element makes depicting network models especially tricky because no sooner than you construct a network, it has changed and as this applies to Julius Caesar we will explore several techniques that will help you properly account for time in your own network mode.

Using DNA, our aim is to discover what Julius Caesar could not discern for himself; how he was vulnerable in his own empire by failing to understand the complex multi-modal evolving network around him. In doing so, we will introduce and explore the power of Dynamic Network Analysis. We aim to show how based on our knowledge of the Julius Caesar network as presented by Shakespeare; a DNA analyst could have made certain recommendations, based on rock-solid mathematical computations, to Caesar, which might have seen him carrying on his rule as emperor of Rome and conquering the rest of known world as he probably would have liked to have done.

Although a skeptic might conjecture that he too would have ignored our insights, much as he did the dire warnings of the Soothsayer, the nightmares of his wife Calpurnia and the advice of Artemidorus

shortly before his ill-fated trip to the Roman Senate. But, that only underscores a human volatility that can in part affect the impact of a well put together Network Model: is the person who is looking at the model shrewd enough to see what it really is?

We know one thing: if we presented Julius Caesar our findings, based on the authority of applying the cutting-edge computational mathematics of Dynamic Network Analysis, Decius Brutus would have had a much harder time convincing Julius Caesar that his wife's dream was merely misinterpreted and that he *should* attend the Senate meeting that day, where he would promptly be stabbed by his closest friends. Our findings would have given him much pause. Still, Caesar would have to act upon them by making some kind of policy decision. He seemed to "go it alone" in that regards and it cost him his life.

Nonetheless, rooted in the knowledge of DNA, a better policy for Julius Caesar could have been crafted for sure to avert his impending doom. We can take solace in our lesson, however, that grounded with the results of better analytical methods we might construct policies that would prevent a network from doing the same again be it for nefarious purposes or altruistic. Along those lines, we should add that our purpose is not necessarily to show how DNA can buttress the continuation of a tyrant, as Cassius might have argued, but just demonstrate how useful it can be when applied with foresight and skill. Brutus could have well used his own DNA model, perhaps, to see the likely outcome of killing Caesar. He might not have needed an ill dream to tell him that Rome would be divided in two. He might have only needed his DNA model.

Therefore, before we begin to build our network model of the Julius Caesar world, we first need to explain to Caesar what a network is, what makes them up, what are the best ways of analyzing them and what challenges are faced in analyzing complex multi-modal networks over periods. We begin with the basics for those Roman citizens of network analysis lacking in the rudiments as we are sure Caesar would have been in the same class as the soothsayers and cobblers in that regards.

Measures

Once you have a network, the graph representation of bunch of things and the links between them, you can begin to identify characteristics of that network. One of the first things to consider is that some nodes stand out. Julius Caesar, for example, was connected to more people in ancient Rome than a soothe-sayer.

There are a large number of measures that identify which things in a network are important or key. The set we are concerned with, at least initially, are those that measure the extent to which a node is of central importance.

A measure if an algorithmic function that tells us something insightful about a network. In some ways, dynamic network analysis is built upon the ability to apply measures to a complex network model and draw mathematically robust conclusions from those Measures. Therefore, we are going to introduce and discuss three powerful Measures in this chapter. They are *Degree Centrality, Betweeness and Eigenvector Centrality.*

We are going to learn these three Measures and apply them to a network model of our Julius Caesar network. In this example, we are using as input only a graph, which is a single network of Entities and the ties they share. This graph will be an Agent by Agent, or *Who by Who*, graph. This sort of graph is probably the type most people intuitively construct when performing rudimentary network analysis: It is merely a visual model of *Who* knows *Who*. Now, some Measures, we will learn, can be run across multiple Entity Class networks. So, we are going to get a sneak preview of our Julius Caesar Network below, for the purposes of introducing 3 key Measures that can be run on it. The idea, at this point, is to hint at the methodology that a Dynamic Network Analyst employs to analyze a complex MetaNetwork. We will then understand that a Network is made up of Entities and the Entities fall into Entity Classes.

The Entities can be analyzed with a Measure. What follows is *Who* by *Who* of the Julius Caesar Network. We will learn about and apply the Measures of *Betweeness, Eigenvector Centrality and Degree Centraility* (Carley 2005).

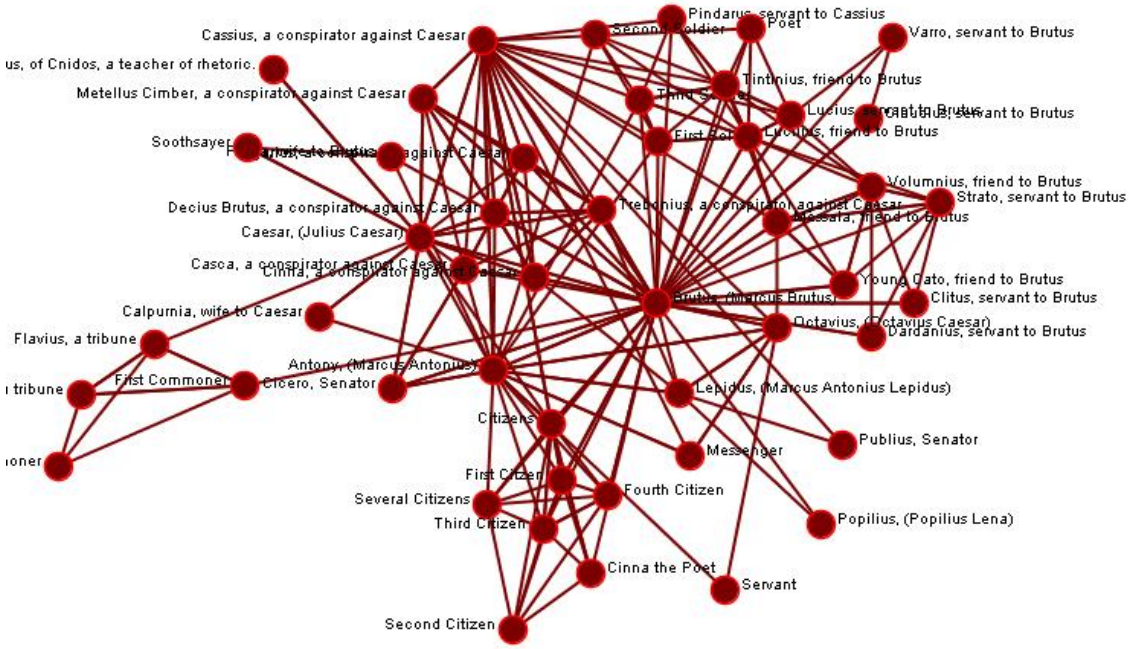


Figure 3: The Julius Caesar *Who* Network. Connections are represented by lines, which are called "Ties". Each person is an Entity.

Degree Centrality

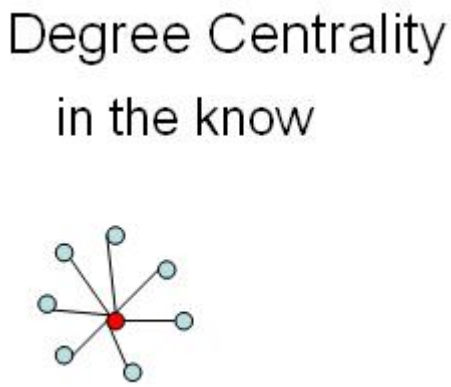


Figure 4: A basic measure to determine how an Entity is connected.

Degree Centrality is a Measure that tells the Dynamic Network Analyst how many other Entities are connected to the Entities we care about. For example, in the model of our Julius Caesar Network, how many people does Julius Caesar talk to or how many other people does he have some sort of meaningful connection with? It is one of the key Measures used by those interested in the analysis of networks.

How you calculate degree centrality depends on whether the network is symmetric (i.e. if A is connected to B is B connected to A?) and whether you want an answer that is normalized (i.e. to lie between 0 and 1 so that you can compare the Degree Centrality across networks of different size).

When the matrix is symmetric we use Total Degree. In this case, the non-normalized version is just:

$$\text{entity i} = \frac{1}{2(n-1)} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} X(i,j)$$

(Wasserman 1994)

This can be normalized by dividing by the N-1. That let's you compare values across networks of different sizes.

When the matrix is not symmetric and direction matters, we can use *Total Degree*, *In-Degree* or *Out-degree*. We use *Total Degree* if we want to know how connected the node is. We use In-Degree if we want to know how many other nodes connect to the node we care about; e.g., how many people tell Caesar something. We use Out-Degree if we want to know how many other nodes, or Entities are connected to by the node we care about; e.g., how many people does Caesar tell something to.

Without getting into the technical details about applying the measure of Degree Centrality in this *Who by Who* network of the Julius Caesar network model, we will run the measure Degree Centrality and discover who is the most connected Entity, the most important *Who* in this network. Will we be surprised?

We will then apply the measures of *Betweeness* and *Eigenvector Centrality* and take a look at the results. First, let us learn something about *Betweeness and Eigenvector Centrality*.

Betweenness

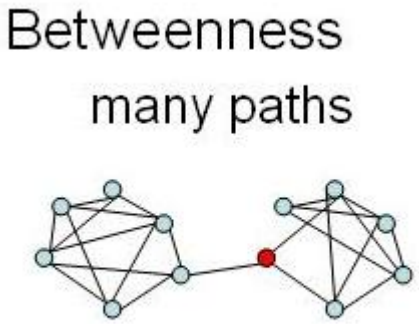


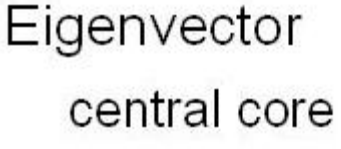
Figure 5: Entity in Red has high "Betweeness" Value. Such Entities tend to be gatekeepers.

Betweeness tells us which node is the most connected to other parts of a network, perhaps groups. For example, which person is central to the Julius Caesar plot network? In terms that are more mathematical Betweeness measures the number of times that connections must pass through a single individual in order to be connected. This measure indicates the extent that an individual is a broker of indirect connections between all others in network, similar to a gatekeeper for information flow in the organization. We can see that in the Julius Caesar network such information would be highly valuable to Caesar. People that occur on many shortest paths between other People have higher Betweeness than those that do not. It is one of the key measures used by those interested in networks. The formula that drives this Measure is:

$$\text{Network Betweenness Cent.} = \left(\sum_{1 \le i \le n} \bar{d} - d_i \right) / (n-1)$$

(Freeman 1979)

Eigenvector Centrality



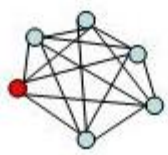


Figure 6: Eigenvector value identifies those who are connected to those that have the most connections.

Eigenvector Centrality tells us which node is the most connected to other parts of a network, perhaps groups. Eigenvector Centrality reflects one's connections to other well-connected people. A person connected to other isolated people has a lower score, even with many connections (i.e. high degree centrality). A person well connected to well connected others can spread information much more quickly than one who only has connections to lesser important people in a network. Depending on their formal role, people with higher scores of Eigenvector Centrality could be important when rapid communication is needed. For instance, if Julius Caesar needs to sound the alarm to his army, he would naturally give the message to someone that was connected to all the other most important people in his empire.

Agent Level Results of applying the Betweeness, Closeness and Eigenvector Centrality measures on the Julius Caesar Network

	Boundary Spanner: Agent x Agent	Boundary Spanner, Potential: Agent x Agent	Capability: Agent x Agent	Centrality, Betweeness: Agent x Agent	Centrality, Bonacich Power: Agent x Agent	Centrality, Closeness: Agent x Agent
Antony, (Marcus Antonius)	0	0.0690474	0.635424	0.0953426	24	0.211712
Artemidorus	0	0	0.00966407	0	2	0.174721
Brutus, (Marcus Brutus)	0	0.1407*	0.9933*	0.4478*	48*	0.229268
Caesar, (Julius Caesar)	1*	0.07943	0.453836	0.121866	23	0.207965
Calpurnia	0	0	0.00966407	0	2	0.174721
Casca	0	0.00165092	0.00966407	0.000886525	1	0.167857
Cassius	0	0.0468244	0.364576	0.0610645	15	0.208889
Cicero, Senator	0	0.0466277	0.0287899	0.0214616	4	0.198312
Cinna the Poet	0	0.0200311	0.0287899	0.00460993	4	0.172794
Cinna	0	0.00777538	0.115369	0.00417529	8	0.2
Citizens	0	0.0350201	0.0826096	0.0308946	7	0.200855
Claudius, servant to Brutus	0	0	0.00966407	0	1	0.189516
Clitus, servant to Brutus	0	0	0.00966407	0	1	0.189516
Dardanius	0	0	0.00966407	0	1	0.189516
Decius Brutus	0	0.00113495	0.0585369	0.000740056	6	0.198312
First Citzen	0	0.0232988	0.0411643	0.00983022	5	0.198312
First Commoner	0	0.136217	0.0200575	0.0313488	3	0.0222222
First Soldier	0	0.00202257	0.0411643	0.000698207	5	0.197479
Flavius, a tribune	0	0.0425889	0.0200575	0.00980133	3	0.0222222
Fourth Citizen	0	0.0629295	0.0826096	0.0217237	7	0.200855
Lepidus, (Marcus Antonius Lepidus)	0	0.0163688	0.0585369	0.00565063	6	0.185039
Ligarius	0	0.000143558	0.0826096	7.70891e-005	7	0.199153
Lucilius	0	0.0286321	0.0826096	0.0186698	7	0.197479
Lucius	0	0.013144	0.0287899	0.00504154	4	0.196653
Marullus	0	0	0.0200575	0	3	0.0222222
Messala	0	0.0455917	0.0826096	0.0209848	7	0.200855

Messenger	0	0.000803925	0.0139358	0.000154178	2	0.17803
Metellus Cimber	0	0.000777128	0.0826096	0.000447117	7	0.199153
Octavius	0	0.0264115	0.0411643	0.0151957	5	0.197479
Pindarus	0	0.016264	0.00966407	0.0043668	1	0.170909
Poet	0	0	0.0411643	0	5	0.2448*
Popilius	0	0	0.00669285	0	0	0.0208333
Portia, wife to Brutus	0	0	0.00669285	0	0	0.0208333
Publius, Senator	0	0	0.00669285	0	0	0.0208333
Second Citizen	0	0	0.00669285	0	0	0.0208333
Second Commoner	0	0	0.00669285	0	0	0.0208333
Second Soldier	0	0	0.00669285	0	0	0.0208333
Servant	0	0	0.0139358	0	2	0.215596
Several Citizens	0	0	0.0585369	0	6	0.2
Soothsayer	0	0.00452208	0.0139358	0.000693802	2	0.17603
Strato	0	0.0017865	0.0585369	0.000616713	6	0.198312
Third Citizen	0	0.0173059	0.0585369	0.00730169	6	0.2
Third Soldier	0	0.0101237	0.0826096	0.00427139	7	0.199153
Tintinius	0	0.0443652	0.158869	0.0255253	9	0.202586
Trebonius	0	0.00656877	0.115369	0.00428322	8	0.2
Varro, servant to Brutus	0	0	0.0200575	0	3	0.236181
Volumnius	0	0.00418711	0.158869	0.00192723	9	0.200855
Young Cato, friend to Brutus	0	0.0477331	0.0411643	0.0146469	5	0.197479
MIN	0	0	0.00669285	0	0	0.0208333
MAX	1	0.140672	0.993307	0.44784	48	0.244792
AVG	0.0208333	0.0208333	0.0914293	0.0206695	5.97917	0.164542
STDDEV	0.142826	0.0322756	0.175839	0.0665964	7.85941	0.0703165

	Centrality, Column Degree: Agent x Agent	Centrality, Eigenvector: Agent x Agent	Centrality, In Degree: Agent x Agent	Centrality, Information: Agent x Agent	Centrality, Inverse Closeness: Agent x Agent	Centrality, Out Degree: Agent x Agent
Antony, (Marcus Antonius)	0.255319	0.0619074	0.255319	0.0377991	0.609929	0.510638
Artemidorus	0.0212766	0.00770619	0.0212766	0.0158618	0.356383	0.0425532
Brutus	0.7447*	0.1046*	0.7447*	0.04092*	0.7553*	1.021*
Caesar	0.361702	0.0675592	0.361702	0.0369802	0.58156	0.489362
Calpurnia	0.0638298	0.0119166	0.0638298	0.0154776	0.356383	0.0425532
Casca	0.276596	0.0446596	0.276596	0.00990439	0.320922	0.0212766
Cassius	0.404255	0.0559228	0.404255	0.0355994	0.578014	0.319149
Cicero, Senator	0.170213	0.0331044	0.170213	0.0206388	0.478723	0.0851064
Cinna the Poet	0.0425532	0.0111581	0.0425532	0.02305	0.368794	0.0851064
Cinna, a conspirator against Caesar	0.12766	0.0254635	0.12766	0.0302842	0.514184	0.170213
Citizens	0.340426	0.0514455	0.340426	0.0279015	0.510638	0.148936

Claudius, servant to Brutus	0.0425532	0.00637031	0.0425532	0.0100655	0.41844	0.0212766
Clitus, servant to Brutus	0.0638298	0.00714599	0.0638298	0.010106	0.41844	0.0212766
Dardanius, servant to Brutus	0.0638298	0.00714599	0.0638298	0.010106	0.41844	0.0212766
Decius Brutus	0.234043	0.0428735	0.234043	0.0273288	0.492908	0.12766
First Citizen	0.12766	0.0161793	0.12766	0.025085	0.485816	0.106383
First Commoner	0.0638298	0.00628556	0.0638298	0.0163656	0.0638298	0.0638298
First Soldier	0.0851064	0.0157779	0.0851064	0.0252988	0.482269	0.106383
Flavius, a tribune	0.0638298	0.0042845	0.0638298	0.0163858	0.0638298	0.0638298
Fourth Citizen	0.0425532	0.0155773	0.0425532	0.0290455	0.510638	0.148936
Lepidus	0.0638298	0.0160873	0.0638298	0.0275709	0.441489	0.12766
Ligarius	0.148936	0.0379348	0.148936	0.0290745	0.503546	0.148936
Lucilius, friend to Brutus	0.212766	0.0141151	0.212766	0.0280352	0.496454	0.148936
Lucius, servant to Brutus	0.12766	0.0125802	0.12766	0.0230115	0.471631	0.0851064
Marullus, a tribune	0.0425532	0.000639301	0.0425532	0.0163038	0.0638298	0.0638298
Messala, friend to Brutus	0.106383	0.0145146	0.106383	0.0286805	0.510638	0.148936
Messenger	0.0638298	0.00867296	0.0638298	0.014964	0.37766	0.0425532
Metellus Cimber,	0.170213	0.0379348	0.170213	0.028992	0.503546	0.148936
Octavius	0.212766	0.0342393	0.212766	0.0243555	0.482269	0.106383
Pindarus	0.12766	0.0188869	0.12766	0.00904854	0.338652	0.0212766
Poet	0	0.0116009	0	0.026136	0.492908	0.106383
Popilius	0.0638298	0.0128541	0.0638298	0	0	0
Portia, wife to Brutus	0.0638298	0.00997098	0.0638298	0	0	0
Publius, Senator	0.0425532	0.00410694	0.0425532	0	0	0
Second Citizen	0.12766	0.00719281	0.12766	0	0	0
Second Commoner	0.0638298	0.000639302	0.0638298	0	0	0
Second Soldier	0.106383	0.00711434	0.106383	0	0	0
Servant	0	0.00548353	0	0.0163087	0.388298	0.0425532
Several Citizens	0.0212766	0.0155773	0.0212766	0.0279114	0.5	0.12766
Soothsayer	0.0425532	0.00827487	0.0425532	0.0158122	0.370567	0.0425532
Strato, servant to Brutus	0.0638298	0.00954996	0.0638298	0.0275562	0.492908	0.12766
Third Citizen	0.106383	0.0161793	0.106383	0.0267394	0.5	0.12766
Third Soldier	0.0851064	0.016797	0.0851064	0.0284707	0.503546	0.148936
Tintinius	0.12766	0.0161435	0.12766	0.0295441	0.531915	0.191489
Trebonius	0.191489	0.0388523	0.191489	0.0299904	0.514184	0.170213
Varro, servant to Brutus	0	0.00704908	0	0.0206449	0.453901	0.0638298
Volumnius,	0.0638298	0.0110998	0.0638298	0.0312557	0.524823	0.191489
Young Cato, friend to Brutus	0.0638298	0.00877883	0.0638298	0.0253888	0.482269	0.106383
MIN	0	0.000639301	0	0	0	0
MAX	0.744681	0.104646	0.744681	0.0409208	0.755319	1.02128
AVG	0.127216	0.0208333	0.127216	0.0208333	0.390219	0.127216
STDDEV	0.12931	0.0203971	0.12931	0.0108787	0.192218	0.167222

Chapter 2: The Meta-Network

Understanding the Julius Caesar network more dynamically means having a more thorough understanding of the MetaNetwork. What does that mean exactly? It means understanding that networks are not just about people. Networks are ways of connecting the who , what, where, why, how and when. Each type of who, what, where, why, how and when is called an entity class. The items within them are called nodes or entities. The set of networks for three or more of these entity classes is called the meta-network. The entire field of Dynamic Network Analysis is based on the concept of the MetaNetwork, which in turn is based on graph theory (more on that later). Therefore, we need to ask what is a Meta-Network and how does it relate to our Julius Caesar dataset?

So, what is an *Entity*? Essentially an Entity is what we are analyzing; it is something that will obviously be of particular network importance to us for one reason or another and the basis for the conduct of our dynamic network analysis. We call anything we are interested in analyzing an Entity. An "Entity" can literally be anything you can think of; on an abstract level, an Entity may be merely a dot in a visual network model. It is what we are networking and connecting our ties to and thus looking at in detail visually. In terms of all things under the sun, an Entity literally that and the sun.

Entities

An "Entity" is essentially the building block of all networks. An Entity is a dot or node in network. It is what we are networking. It is what we are looking at and it can literally be anything that you can possibly think of in terms of what you can possibly imagine. You name it – it is an entity! We can't build a network without *Entities*. In Dynamic Network Analysis an entity is often best described as a *Who, What, Where, When, How or Why*. These descriptions are the hallmark of any good news story and are convenient ways to describe any complex system or story. Networks tell stories and vice versa. We can almost extrapolate any network from a story and a network model may indeed tell a story, but we are getting ahead of ourselves.

If you take a few moments to ponder this, most anything you can think of can fit into one of these categories. So, think of Entities in terms of *Whos, Whats, Wheres, Whens, and Whys*.:

- A *Who* such as Julius Caesar, Cassius, Brutus, CEOs, famous historical people, imaginary people, myths, your friends, family members, terrorists, people that owe you money, scientists, celebrities, athletes, religious figures, etc.
- A *Where* such as The Roman Senate, The streets of Rome, Brutus' House, Planets, cities, galaxies, stores, swimming pools, lakes, rivers, oceans, countries, roads, etc.
- A *What* such as a dagger, computers, satellites, cars, cell phones, food, money, emails, molecules, etc.
- A *Why* Julius Caesar is becoming too powerful and so should be killed, other beliefs and attitudes.

The Entities in Julius Caesar

In the following list, we have the characters that make up our Julius Caesar network, which as we have discussed is based on William Shakespeare's *Tragedy of Julius Caesar*. All of the characters on this list constitute an Entity, which for the purposes of Dynamic Network Analysis we say is a *Who*. After all, they are people, although in the fictional sense. They are, for our consideration, People. They are the *Whos*. Other Entity Classes, which you will see, allow us to put certain Entities into other different containers, which, perhaps you have guessed by now, correspond to the *Who, What, When, Where and Why* model. There are even other components as well, but we will explore those more deeply as we learn about Entity Classes soon enough. For now, let us visit our *Whos* as they relate to the Julius Caesar network model we are going to build in the next chapter. Here are the *Whos*:

Cast of Characters in Julius Caesar (*Whos*)

Antony, (Marcus Antonius)
Artemidorus, of Cnidos, a teacher of rhetoric.
Brutus, (Marcus Brutus)
Caesar, (Julius Caesar)
Calpurnia, wife to Caesar
Casca, a conspirator against Caesar
Cassius, a conspirator against Caesar
Cicero, Senator
Cinna, a conspirator against Caesar
Cinna the Poet
Citizens
Claudius, servant to Brutus
Clitus, servant to Brutus
Dardanius, servant to Brutus
Decius Brutus, a conspirator against Caesar
First Citzen
First Commoner
First Soldier
Flavius, a tribune
Fourth Citizen
Lepidus, (Marcus Antonius Lepidus)
Ligarius, a conspirator against Caesar
Lucilius, friend to Brutus
Lucius, servant to Brutus
Marullus, a tribune
Messala, friend to Brutus
Messenger
Metellus Cimber, a conspirator against Caesar
Octavius, (Octavius Caesar)
Pindarus, servant to Cassius
Poet
Popilius, (Popilius Lena)
Portia, wife to Brutus
Publius, Senator
Second Citizen
Second Commoner
Second Soldier
Servant
Several Citizens
Soothsayer
Strato, servant to Brutus
Third Citizen
Third Soldier
Tintinius, friend to Brutus
Trebonius, a conspirator against Caesar
Varro, servant to Brutus
Volumnius, friend to Brutus
Young Cato, friend to Brutus

So now that we met the *Whos* in Julius Caesar, remember that an Entity can be literally anything. But for a moment, think of all the networks that would compose a *MetaNetwork* called "Existence." You couldn't describe "Existence" with one type of Entity. If indeed it were possible at all, it is plainly obvious that we would need to interconnect all sorts of different types of Entities. We would need a model beyond a mere network of same type Entities. We would need a better model, one that would incorporate different Entity classes and allow us to perform an analysis on the model that way. Such a model is called a *MetaNetwork*.

We haven't talked about the MetaNetwork yet, but we will in the next chapter. Right now, the idea is to get a firm understanding of what an Entity is and how Entities are the building blocks of

networks. In the next chapter, we want to create a MetaNetwork of Julius Caesar, which will attempt to capture and present to us a model of all the Entities that make up the plot of Shakespeare's classic play. Even though it may seem small, with only 48 *Whos*, just watch and see what happens when we start factoring in locations, knowledge bases, events – every Entity-types that can meaningfully describe a *Who, What, When, Where and Why* of the Julius Caesar network model. Indeed, things can get complex fast as you will also come to see.

Entity Classes

It is useful to classify certain type of *Entities* into categories or classes. A group of entities of the same type is referred to as an *Entity Class*. The relation of Entity Class to Entities is sort of like Genus to Species in the Linnaeus system of classification. Toward those ends, there are 6 major entity classes that social scientist have determined to be of the most value in network analysis: *Agents, Locations, Events, Resources, Tasks, Knowledge*. These are the Genus categories of which we can neatly tuck most species – Entities.

There are six major entity classes that social scientist have determined to be of virtual suigeneris value when it comes to network analysis: *Agents, Locations, Events, Resources, Tasks, Knowledge*. These are the Genus categories of which we can neatly tuck most species – Entities. Are they beginning to sound familiar now? Can you see the parallel between these types and the *Who, What, When, Where, Why and How* model?

These classifications are what drive many of the advanced mathematical algorithms that can make a highly complex MetaNetwork comprehensible to an analyst. The good news is that nearly anything that can be an Entity can neatly fall into one of these Entity Classes and you will see that as we build the Julius Caesar model, we will place most of our Entities into such containers as the Entity Class types described above to arrive at our Who, What, Where, When and How.

Upon revisiting our Entity definition we can extrapolate Entity Classes into the traditional *Who, What, When, Where, Model*: An "Agent" could be any person a Who: Roman soldier, Soothsayer, Calpurnia, Cicero, historical figure, family member, terrorist, children, teachers, etc.; A "Resource" could literally be a dagger, poison, a cloak, a crown, a short sword, a computer, money, bombs, tools, books, ingredients; an "Event" could be 9-11, the JFK Assignation, The Super Bowl, A wedding, a funeral, an inauguration, etc.; a "Task" could be planning, executing, balancing a check book, conducting a symphony, overhauling a transmission, etc.; a "Knowledge" could be the science of DNA, Trigonometry, History, English, Economics, etc.; a "Location" could be London, The Middle East, Earth, Outer Space, Mars, The Sun, etc.

So when building a network the DNA scientist needs to be aware of how to categorize the Entities he or she wishes to study, that is a decision has to be made as to what Entity Class it makes sense to put the Entity in. To a certain extent one can make up their own Entity Classes sole and separate from the ones aforementioned. However, it probably is not wise to do so; by fitting your Entities into a traditional, if not obvious, Entity Class, we can then run powerful Measures on them. What is a Measure? We will get to that shortly.

The major Entity Classes of *Agents, Locations, Knowledge, Resources, Tasks*, and *Events* seem to easily capture most Entities. Some of these, as you will see when building our MetaNetwork of Julius Caesar, will overlap. Unless you have good reason or are intent on blazing your own path in the science of DNA you would be best advised to adhere to the component Entity Classes for your respective model. This goes for our Julius Caesar model and should go for yours as well (Carley 2003). Some of these Entity Classes, as you will learn when building a MetaNetwork, overlap.

The Meta-Network

A Meta-Network is network composed of multiple networks. A MetaNetwork is the sum of all networks we are considering. The Meta-Network is the foundation for Dynamic Network Analysis; it is the resultant model from which we can harness the power of computational mathematics to reveal structure in networks that would otherwise be impossible to reveal, at least to the unaided human mind. This is because multimodal networks, that is networks that make use of several entities, as in Who, What, When, Where, and How, get complex in a hurry (Carley 2005).

These classifications are what drive many of the advanced mathematical algorithms that can make sense out of a complex MetaNetwork incomprehensible to the computer-unassisted mind. The good news for you is that nearly anything that can be an entity can neatly fall into one of these Entity Classes and you will see that as we build the Julius Caesar model.

Armed with the concepts of the MetaNetwork, Entities and Entity Classes, you should now have a firm grasp of the concepts behind The MetaNetwork, how to build one and why it is important. It is our hope that by building your own networks by assigning relationships to your own "various" entities you will see that the building blocks of Dynamic Network Analysis are derived from observational data readily available to anyone as we will show by pulling such information from The Tragedy of Julius Caesar. After all, we are part of networks, some big, some small, some apparent and some invisible to our everyday comprehension.

Moving forward we are going to get into more advanced uses of the MetaNetwork, namely "extending the MetaNetwork" as well as introducing more complex facets of what DNA scientists can do with a MetaNetwork. Here we will introduce the concept of how time affects networks and how networks in turn behave in probabilistic fashion when certain changes occur in your MetaNetwork over time. We will also introduce more complexity by given attributes to our entities.

Naturally, we need to think up of a least two different networks. But, to really hammer the point home, let us come up with three, maybe five different networks relating to Julius Caesar and add them together to see what we get.

Building the Julius Caesar MetaNetwork

Therefore, we will start by constructing a simple social network as will most network models probably follow. This social network will be our first network and it will tell us *Who* knows *Who* in our network. It will be a Who by Who graph, represented visually. This is oftentimes called a social network.

A social network should be something familiar to anyone. Whomever you regularly talk with can be a social network. It can be your friends, family, people you work with or any combination of them. In the case of Julius Caesar the *Who* network is everyone in the network – the play so to speak. We can look no further than the cast of characters from Julius Caesar to discover exactly who the *Whos* are:

Cast of Characters in Julius Caesar (*Whos*)

Antony, (Marcus Antonius)
Artemidorus, of Cnidos, a teacher of rhetoric.
Brutus, (Marcus Brutus)
Caesar, (Julius Caesar)
Calpurnia, wife to Caesar
Casca, a conspirator against Caesar
Cassius, a conspirator against Caesar

Cicero, Senator
Cinna, a conspirator against Caesar
Cinna the Poet
Citizens
Claudius, servant to Brutus
Clitus, servant to Brutus
Dardanius, servant to Brutus
Decius Brutus, a conspirator against Caesar
First Citzen
First Commoner
First Soldier
Flavius, a tribune
Fourth Citizen
Lepidus, (Marcus Antonius Lepidus)
Ligarius, a conspirator against Caesar
Lucilius, friend to Brutus
Lucius, servant to Brutus
Marullus, a tribune
Messala, friend to Brutus
Messenger
Metellus Cimber, a conspirator against Caesar
Octavius, (Octavius Caesar)
Pindarus, servant to Cassius
Poet
Popilius, (Popilius Lena)
Portia, wife to Brutus
Publius, Senator
Second Citizen
Second Commoner
Second Soldier
Servant
Several Citizens
Soothsayer
Strato, servant to Brutus
Third Citizen
Third Soldier
Tintinius, friend to Brutus
Trebonius, a conspirator against Caesar
Varro, servant to Brutus
Volumnius, friend to Brutus
Young Cato, friend to Brutus

From the previous chapter, we know that a network is made up of two things that share a connection. So in creating our social network of Julius Caesar we can say that a character in the play and other character's they know - an acquainted character – together constitute a social network.

So there we have all these characters and now we need to figure out *Who* knows *Who*. In our case, we are going to consider that they know each other if they appear in the same scene. Naturally, this

requires a little close observation our part. We might summarily dismiss merely casual associations in the same scene. This underscores that building social networks can be an art in its own right. After all, you need to establish criteria for association in the social network and that takes human judgment. If you can't exude good human judgment in what constitutes a relationship in a network, chances are this will be reflected in a highly complex MetaNetwork that will only magnify misapplied human judgment. As a loose rule of thumb, in our model, we will say if characters appear in the same scene it could be said they are networked socially and therefore constitute a path between them. Now, we need to build a graph representing such relationships. How do we do this? We binarize relationships. What does that mean?

We create a square graph, meaning we have a row of all our characters and column of all our characters. It would look like such – at least a portion of it:

	Antony, (...	Artemidorus,...	Brutus, (Marc...
Antony, (Marcus Antonius)	0.0	0.0	2.0
Artemidorus, of Cnidos, a teacher of rhetoric.	0.0	0.0	0.0
Brutus, (Marcus Brutus)	1.0	0.0	0.0
Caesar, (Julius Caesar)	0.0	1.0	3.0
Calpurnia, wife to Caesar	0.0	0.0	0.0
Casca, a conspirator against Caesar	0.0	0.0	0.0
Cassius, a conspirator against Caesar	1.0	0.0	4.0
Cicero, Senator	1.0	0.0	1.0
Cinna the Poet	0.0	0.0	0.0
Cinna, a conspirator against Caesar	0.0	0.0	1.0
Citizens	1.0	0.0	1.0
Claudius, servant to Brutus	0.0	0.0	1.0
Clitus, servant to Brutus	0.0	0.0	1.0
Dardanius, servant to Brutus	0.0	0.0	1.0

Figure 7: Social Graph of Julius Caesar

Note that in this example we have given some relationships a higher value than "1". This is called weighting a relationship, but for our purposes, consider any correlating cells with numbers, to contain a "1" as in having a connection. We will get into weighted ties later on.

Once we make all our connections, let us visualize a social network and proceed by representing the above graph in terms of nodes and ties. A node represents a *Who* and a *tie* is a connection between the *Whos*. In a sense, this tells us *Who* knows *Who*.

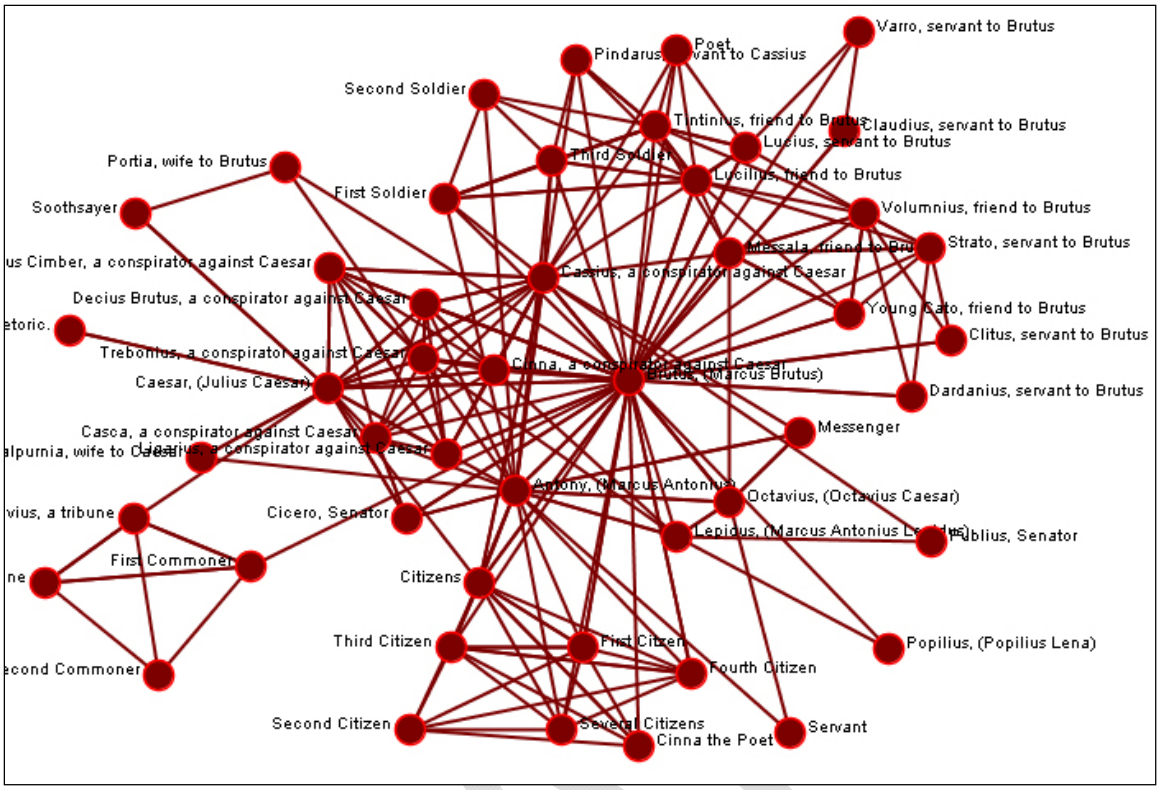


Figure 8: The Whos in Julius Caesar (Who by Who, i.e., Agent x Agent)

We can see that the characters are each represented by a large red dot or solid circle. At this point, we are still keeping it simple. You can also see visually now how just this cast of characters can make for a complex network of relationships.

Again, before we get into truly Dynamic Network Analysis, our goal is to create multiple networks to create our multi-modal "MetaNetwork." So, let's now create another network by repeating the process for building our *Who* by *Who* graph and then another until we are satisfied that we have captured enough pertinent and interesting data that could be of use to us if presented as a MetaNetwork. For purposes of our Julius Caesar model we are going to capture about as many useful networks as possible, graphing the relationships of *Who, What, When, Where, How.*

In our second network, let us decide to build a network out of the places *Where* the characters in Julius Caesar have been seen. This is our *Where* x *Who* network or simply put: *Who* has been *Where*. For this network, we will use the locations of *places* the characters generally have been shown to appear in the play. Here then is the list of locations which will be the foundation of our *Where's* network:

Locations in Julius Caesar	
Battle Tents	Funeral site
Battlefields	Parade to Senate
Brutus' House	Pompey Parade
	Senate
	Streets of Rome

At least to us, these locations can generally be thought to constitute the main scene settings for the characters, without splitting hairs about tents and theatrical asides as conspirators whisper in each other's

ears. Next, we graph our locations against our Cast of Characters, *binarizing* ties as "1s" or "Os" for not ties. In this case, we are giving a "1" if a character had been in that location. Here is a sample of what the graph looks like as we begin to binarize relationships on a graph.

	Battle Tents	Battlefields	Brutus' House	Funeral site	Parade to Se...	Pompey Parade	Senate	Streets of R...
Antony, (Ma...	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0
Artemidorus,...	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0
Brutus, (Mar...	1.0	1.0	0.0	1.0	0.0	1.0	1.0	0.0
Caesar, (Juli...	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0
Calpurnia, wi...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
Casca, a con...	0.0	0.0	1.0	0.0	1.0	1.0	1.0	1.0
Cassius, a co...	1.0	1.0	1.0	0.0	1.0	1.0	1.0	0.0
Cicero, Senator	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0
Cinna the Poet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cinna, a con...	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
Citizens	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0
Claudius, ser...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Clitus, serva...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 9: Graphing *Whos x Where*

Once we have established all our ties as binary code "1s" or "Os" we can represent the graph visually as such:

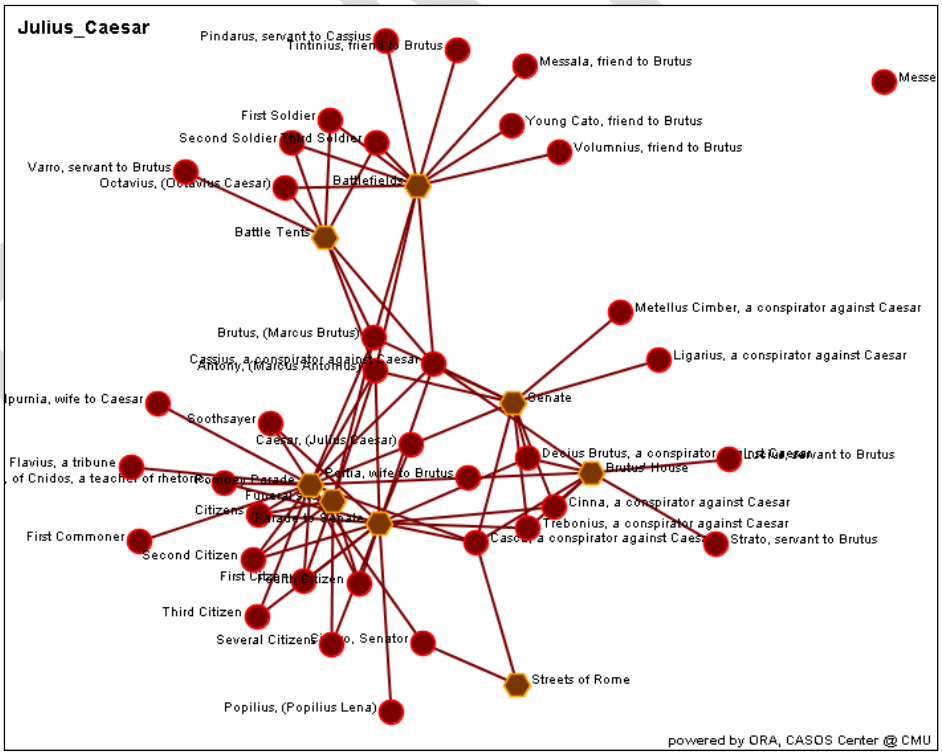


Figure 10: Who by Where (Agent x Location)

Notice how the agents, represented by dots, are connected to the hexagons, which represent the locations. Let us look at some of this *Who* x *Where* network in greater detail.

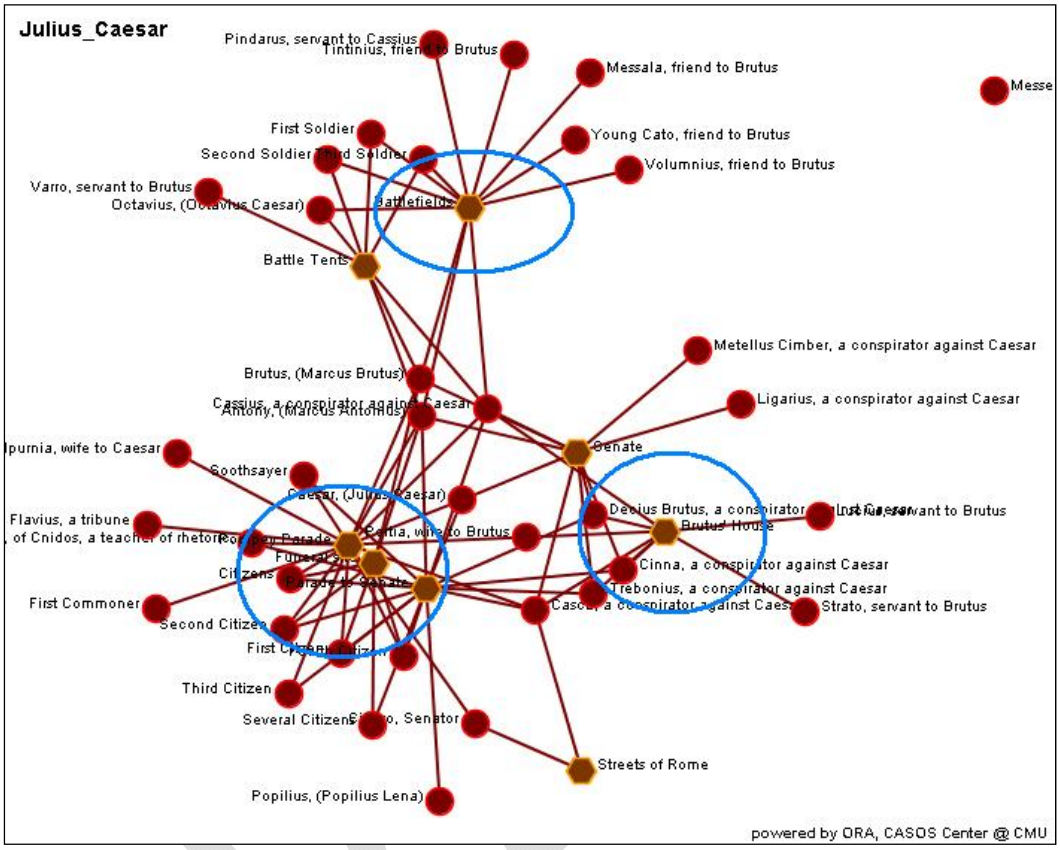


Figure 11: *Where by Location* detail - blue ellipses focus on the location nodes which are orange. Look at the connections the *Whos* have to each location. This is a multimodal network because it visualizing a network of separate Entities (*Whos and Wheres*). By contrast, a *Who* x *Who* network is not multimodal as is a Where by Where.

In the next example, we are going to build yet another network. This network will again take into account multimodal data. To build our next network we are going to graph *Who x What*. The *What* in this case represents Knowledge (K). So, our *Who* by What graph to the Dynamic Network Analysis analyst constitutes an Agent (A) by (K) Knowledge graph. Let us begin by determining a list of what knowledge bases would seem applicable to our graph. Based on our understanding of the play, we have determined the following knowledge sets to be of relevance to the network analyst.

Knowledge set for Julius Caesar – our *What*

Administration
Citizenry
Military
Persuasion
Politics
Prediction

Generally, it our opinion that the knowledge sets displayed by the actors (quite literally) fall into these broad knowledge categories. Perhaps you can think of others? Let us begin to binarize this graph so that we may look at it visually. You will also see that some knowledge bases can be thought of as resources as well, which is why the distinction in some algorithms allows for both R and K input variables.

	Administration	Citizenry	Military	Persuasion	Politics	Prediction
Antony, (Ma...	0.0	4.0	3.0	4.0	3.0	0.0
Artemidorus,...	0.0	0.0	0.0	1.0	0.0	1.0
Brutus, (Mar...	0.0	2.0	2.0	3.0	2.0	0.0
Caesar, (Juli...	0.0	0.0	1.0	0.0	1.0	0.0
Calpurnia, wi...	0.0	0.0	0.0	1.0	0.0	3.0
Casca, a con...	0.0	0.0	0.0	0.0	1.0	0.0
Cassius, a co...	0.0	0.0	0.0	2.0	4.0	3.0
Cicero, Senator	0.0	3.0	0.0	0.0	1.0	0.0
Cinna the Poet	0.0	1.0	0.0	1.0	0.0	0.0
Cinna, a con...	0.0	0.0	0.0	0.0	1.0	0.0
Citizens	0.0	1.0	0.0	0.0	0.0	0.0
Claudius, ser...	0.0	0.0	0.0	1.0	0.0	0.0
Clitus, serva...	1.0	0.0	0.0	0.0	0.0	0.0
Dardanius, s...	1.0	0.0	0.0	0.0	0.0	0.0
Decius Brutu...	0.0	0.0	0.0	3.0	1.0	0.0
First Citzen	0.0	1.0	0.0	0.0	0.0	0.0
First Commoner	0.0	1.0	0.0	0.0	0.0	0.0
First Soldier	0.0	0.0	1.0	0.0	0.0	0.0

Figure 12: Binarize graph Who by What

As we mentioned previously, we simply are creating ties by binarizing our data. However, you will be begin to see which weighted values can impact a graph. There surely is something to be said about Antony having a value of 4.0 when it comes to his knowledge of Persuasion whereas Brutus only has a 3.0. This would seem to make sense when analyzing the dynamics of the Julius Caesar play. In our judgment, Brutus failed to motivate the Citizenry whereas Antony was able to do so because he was more skilled – thus we attributed to him a higher value. Let us visualize this graph and take a look.

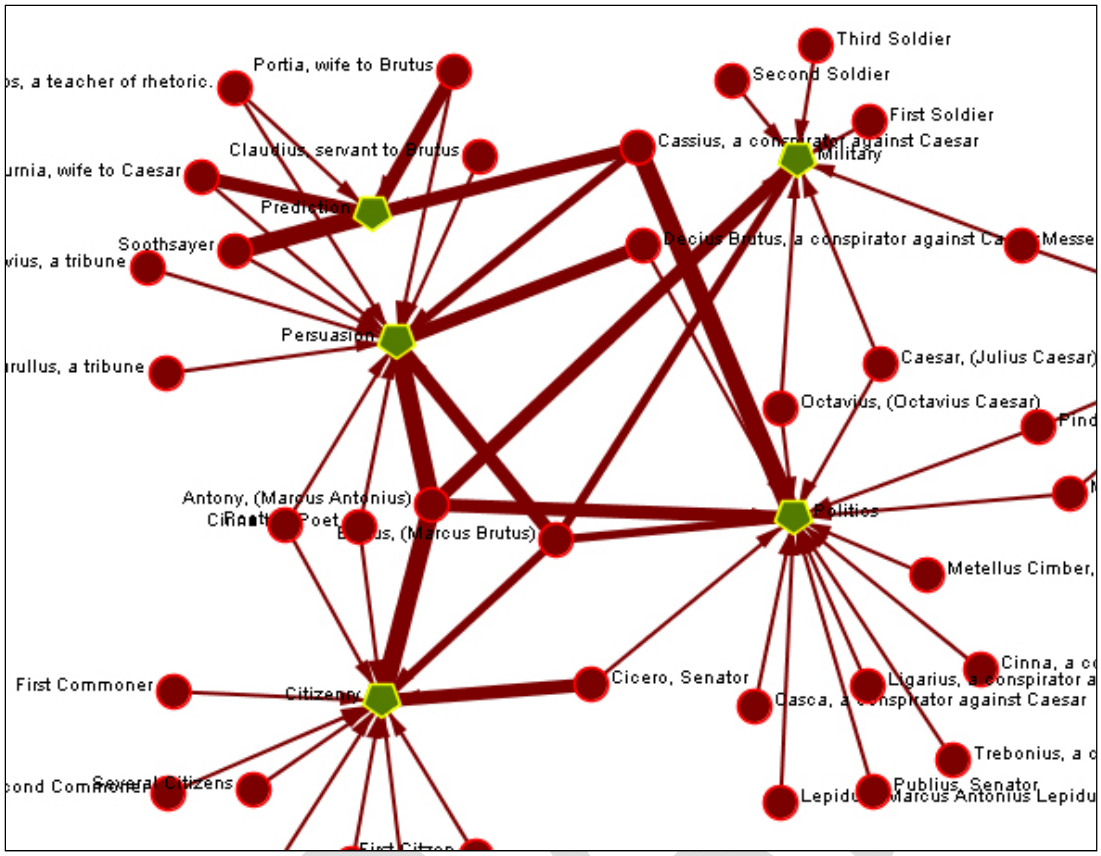


Figure 13: *Where by What.* Agent x Knowledge network. Notice the weighted ties. *Who* seems to have more meaningful connections? Brutus or Antony?

Okay, so far so good. We are compiling graphs and for what ends? We know that individually each graph is telling us something unique and hopefully interesting about the Julius Caesar network. So, towards those ends, let's build a few more multimodal graphs. Once we believe have enough, we will then add the graphs together.

Now let us create *Who* x *How*, or *Agent* (A) by Task (T) multimode network. Just like in the previous examples, here is what we have determined to be the tasks as we took them to be by studying the plot of Julius Caesar in detail. We will skip a list of our tasks and jump right to our binarized graph.

	Title
1	Achieve Victory
2	Attend Senate
3	Avenge Caesar
4	Celebrate Victory
5	Deceive Brutus
6	Defeat Antony/Octavius
7	Defeat Brutus/Cassius
8	Expunge Conspirators
9	Form Coallition
10	Haunt Brutus
11	Justify Murder
12	Kill Caesar
13	Lure Caesar
14	Persaude Citizens
15	Persuade Brutus
16	Persuade Caesar
17	Persuade Citizens
18	Read Will
19	Refute Brutus
20	Support Antony/Octavius
21	Support Brutus/Cassius
22	Warn Brutus
23	Warn Caesar

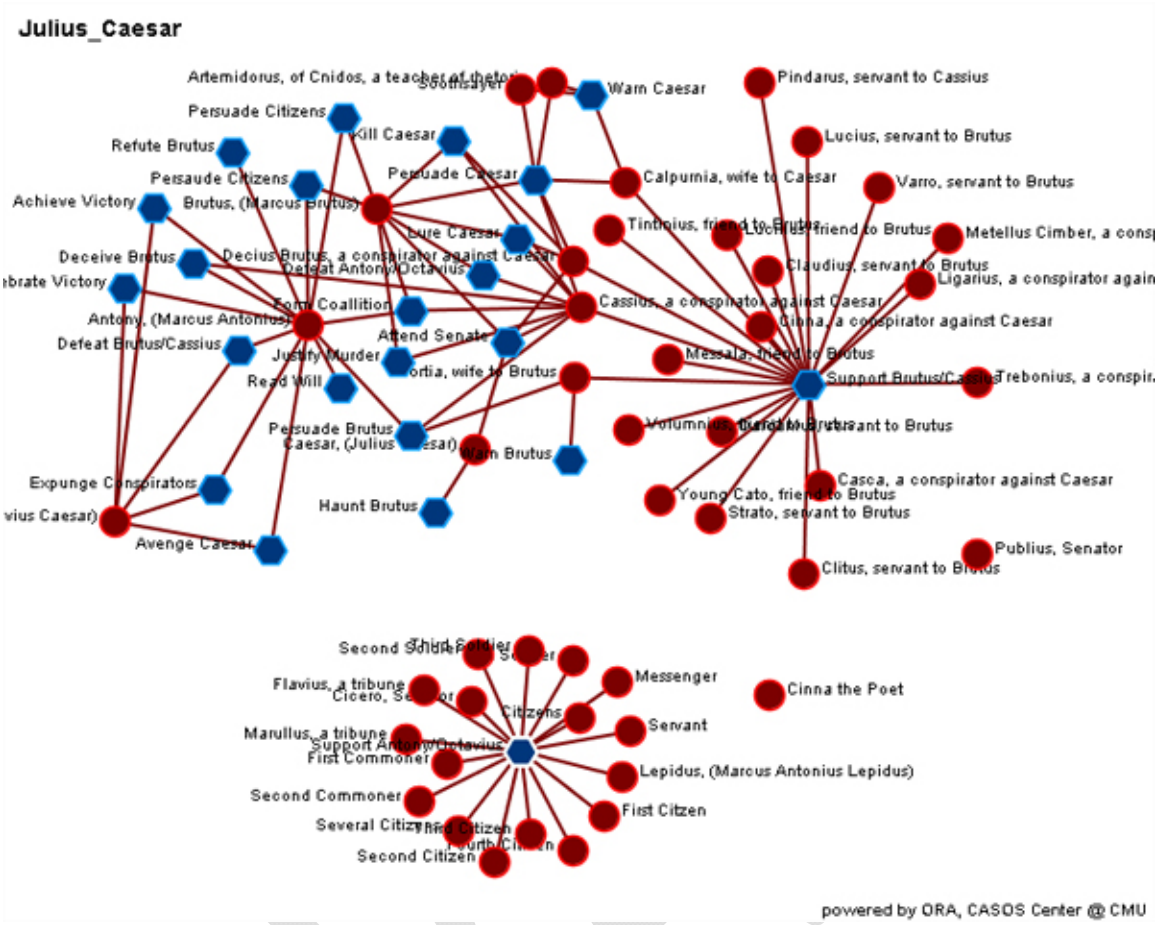
Figure 14: Task list in Julius Caesar

Once again, we have created a list, much like our previous knowledge sets, based on our personal insights into the Julius Caesar network. Let us know look at a binary graph of *Who* is doing *What* task. You can call this a *Who* by *How* or *Who* does *What*. In DNA parlance this is an Agent (A) by Task (T) graph. This later terminology comes in handy when looking at the algorithmic functions we will apply to such graphs.

	Achieve Victory	Attend Senate	Avenge Caesar	Celebrate Vic...	Deceive Brutus	Defea
Antony, (Ma...	☑	☐	☑	☑	☑	☐
Artemidorus,...	☐	☐	☐	☐	☐	☐
Brutus, (Mar...	☐	☑	☐	☐	☐	☑
Caesar, (Juli...	☐	☑	☐	☐	☐	☐
Calpurnia, wi...	☐	☐	☐	☐	☐	☐
Casca, a con...	☐	☐	☐	☐	☐	☐
Cassius, a co...	☐	☑	☐	☐	☑	☑
Cicero, Senator	☐	☐	☐	☐	☐	☐
Cinna the Poet	☐	☐	☐	☐	☐	☐
Cinna, a con...	☐	☐	☐	☐	☐	☐
Citizens	☐	☐	☐	☐	☐	☐
Claudius, ser...	☐	☐	☐	☐	☐	☐
Clitus, serva...	☐	☐	☐	☐	☐	☐
Dardanius, s...	☐	☐	☐	☐	☐	☐
Decius Brutu...	☐	☑	☐	☐	☐	☐
First Citzen	☐	☐	☐	☐	☐	☐
First Commoner	☐	☐	☐	☐	☐	☐
First Soldier	☐	☐	☐	☐	☐	☐
Flavius, a tri...	☐	☐	☐	☐	☐	☐

Figure 15: binary graph Who x How

Let us know look at this binarized data visually.



In this visualization, we can see some interesting relationships about *Who* is doing *What* or *How*. The task shown as the blue hexagon in the lower part of the image is "Support Octavius/Antony". This task is critical to Antony and Octavius ultimately putting down the coalition of Brutus and Cassius. This task according to our model appears to be detached from the rest of the tasks above it. What can draw from that? Perhaps this task was well guarded and there was not a whole lot Brutus and Cassius could do to disrupt the support Antony and Octavius would receive in battle. So, maybe the fault really did lie in the stars?

Are you beginning to notice a pattern on constructing a MetaNetwork? Well, for the purposes of performing a full analysis of the Julius Caesar network, we will proceed to build all of the following graphs following the same methods previously outlined above for each of the aforementioned graphs we will then arrive at graphs, most multimodal, that is using two different types of entities (e.g., agents and events) and a few single mode graphs (e.g., agent x agent, event x event). We put in the parenthetical the lay description of the graph in keeping with our *Who, What, When, Where, How* and *Why metaphor.*

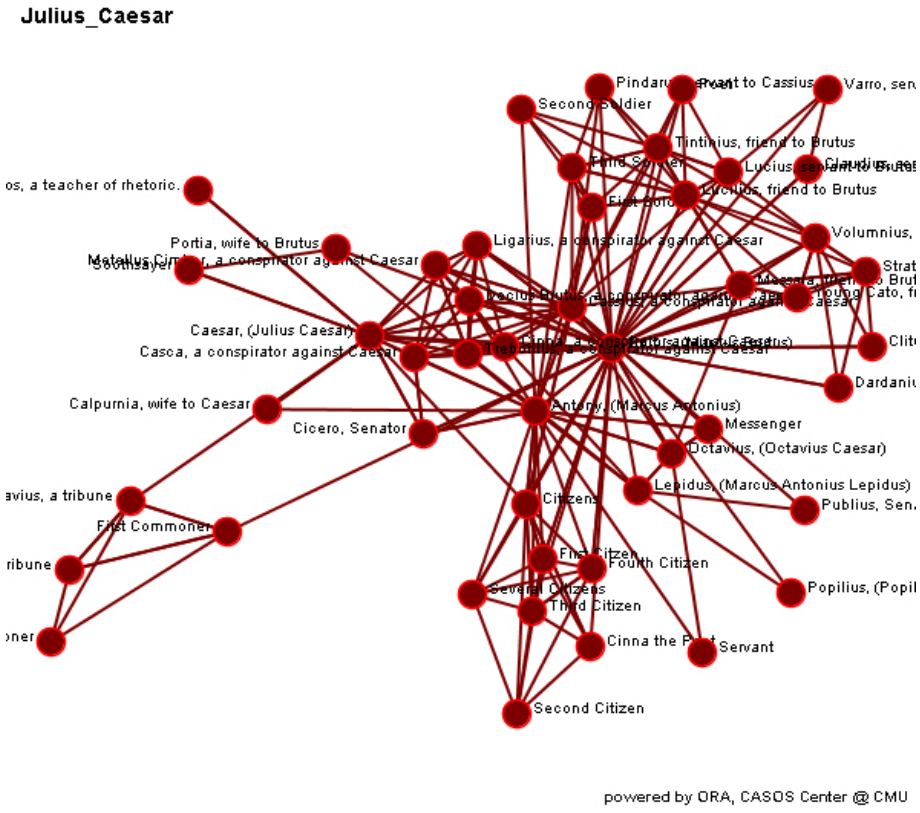
1. Agent x Agent (who knows who)
2. Agent x Event (who goes to what)
3. Agent x Knowledge (who knows what)
4. Agent x Location (who is where)

5. Agent x Task (who does what)
6. Event x Event (When by When)
7. Knowledge x Task (What is needed to do What)
8. Location x Location (where by where)
9. Task x Event (What by When)
10. Task x Task

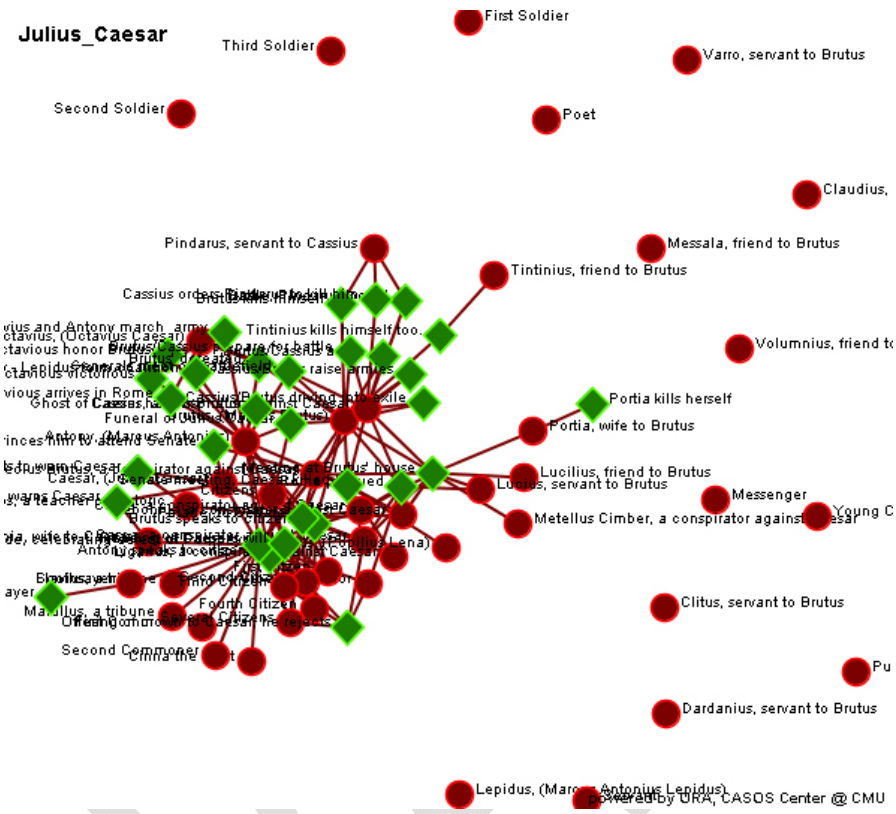
Note that there is one entity set that we are not employing in our Julius Caesar dataset and that is "Resources." Why have we opted not to use resources? Primarily because it seemed that Knowledge (K) was every bit as a Resource (R) and they were in a sense interchangeable in our network. This is a gut call – again human judgment comes into play. You will later discover that many of the same algorithmic measures, which really tell us the most valuable information about our network structure, work on A(K) that is Agent x Knowledge or A(R) Agent x Resources. So there can be an overlap of resources or knowledge or it could be a mere question of semantics. Knowledge can be a resource and resource can be knowledge.

For instance, knowing how to stab someone with a dagger can be a *resource* needed to accomplish the *task* of murdering Caesar or it can be a necessary *knowledge*. In our model, we are going with *knowledge* since *resources* are pretty limited in imperial Rome so far as Shakespeare was concerned – togas, daggers, laurel wreaths, parade pageantry? Now that being said, you could just as easily build a network model using both K and R respectively. You might have computers as a resource (R) and Java as knowledge (K) and the list goes on. Now, in our quest to build a MetaNetwork of the Julius Caesar dataset let us look at the visualizations of all our graphs, which we built each individually just as we did the first ones we discussed earlier in this chapter. At the very end, we will "add" them together to arrive at our MetaNetwork.

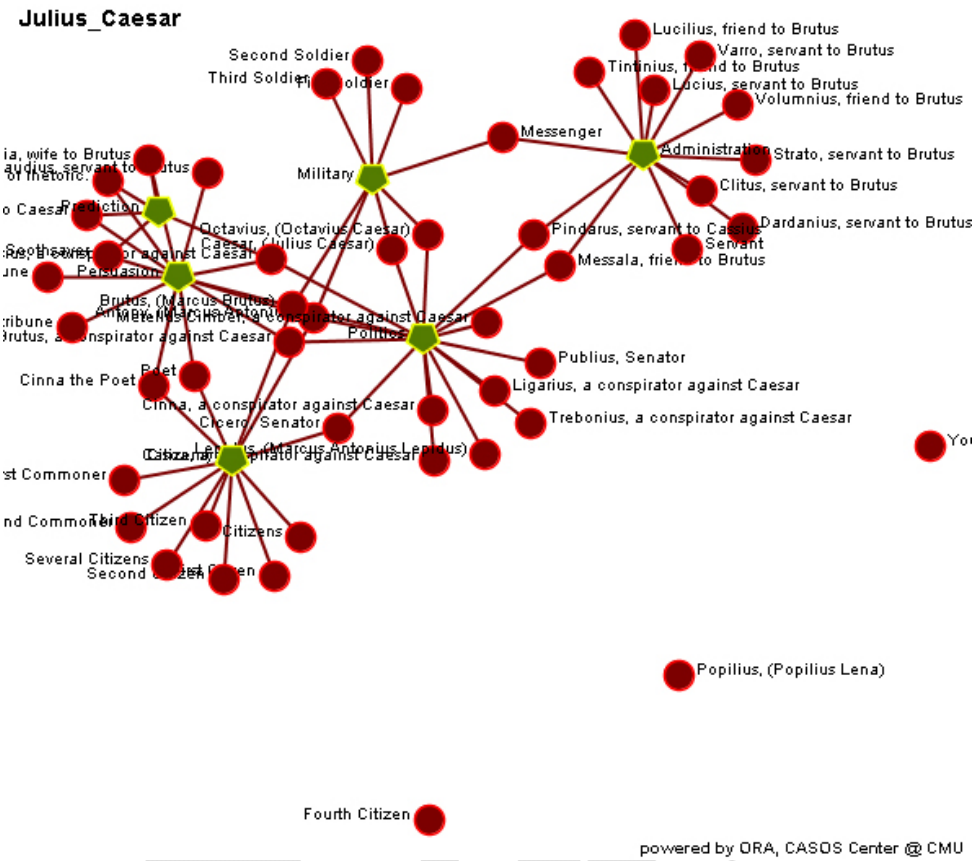
Here then is our Agent x Agent network (*Who* knows *Who*):



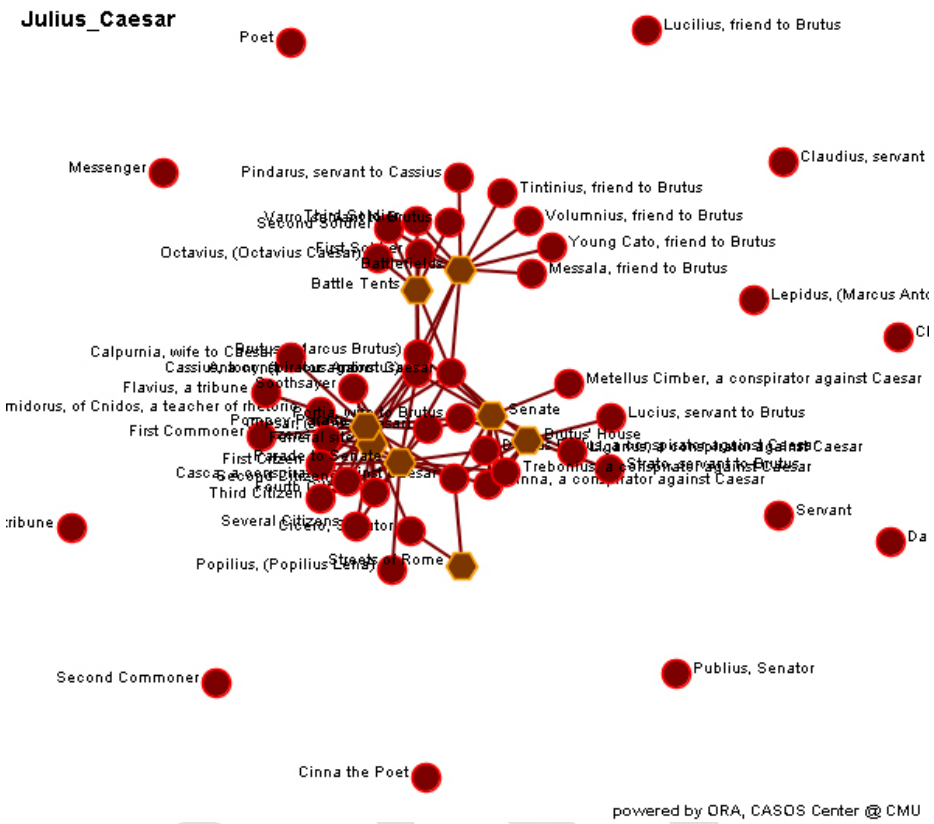
Here is our Agent x Event (*Who* is *Where*):



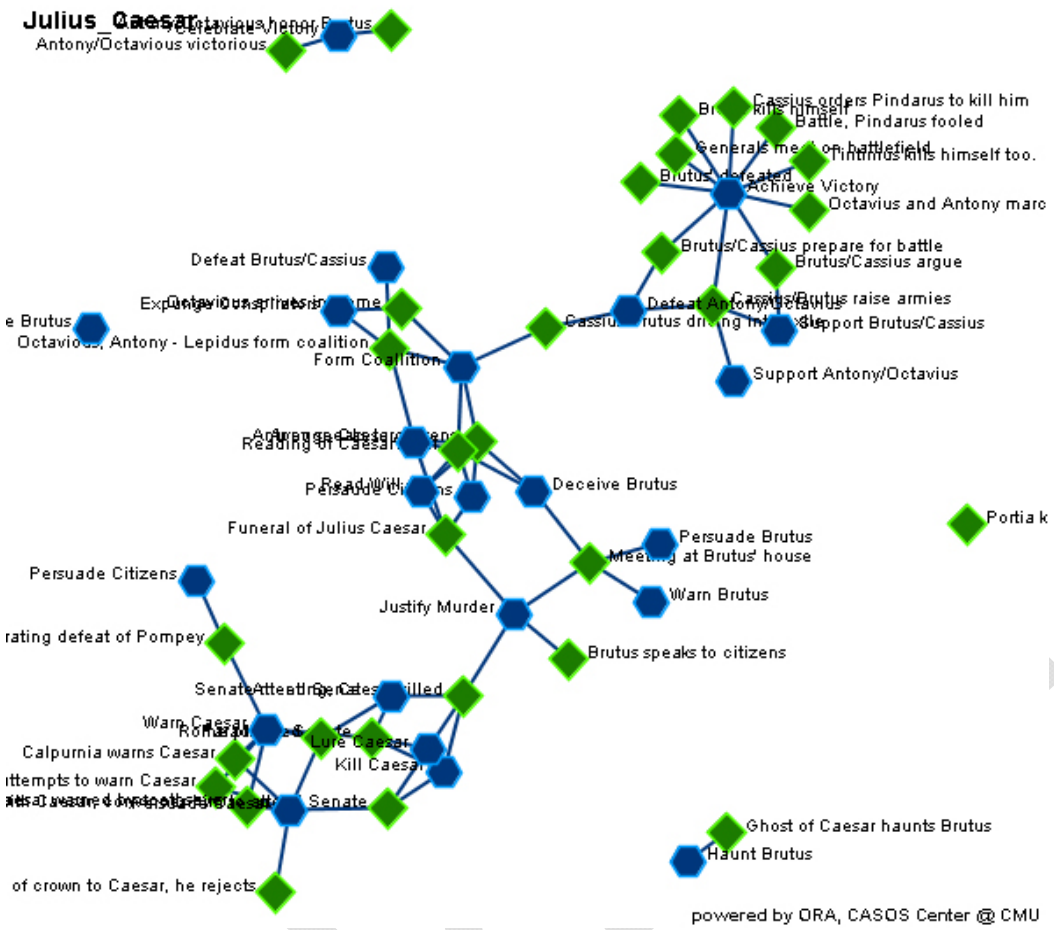
Agent by Knowledge Graph (*Who* knows *What*)



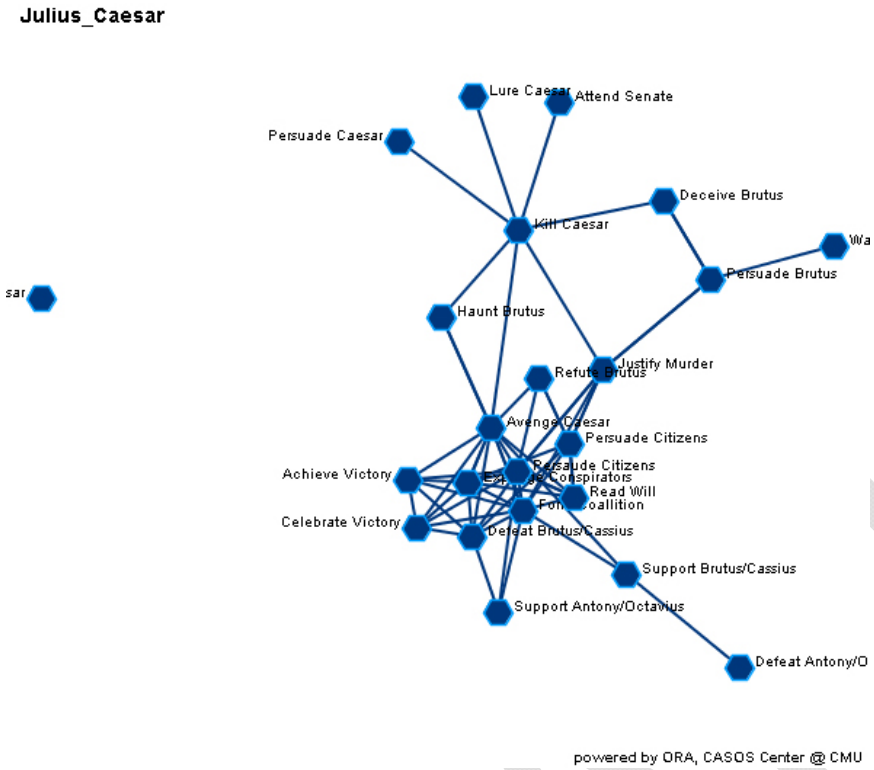
Agent by Location (*Who* was *Where*)

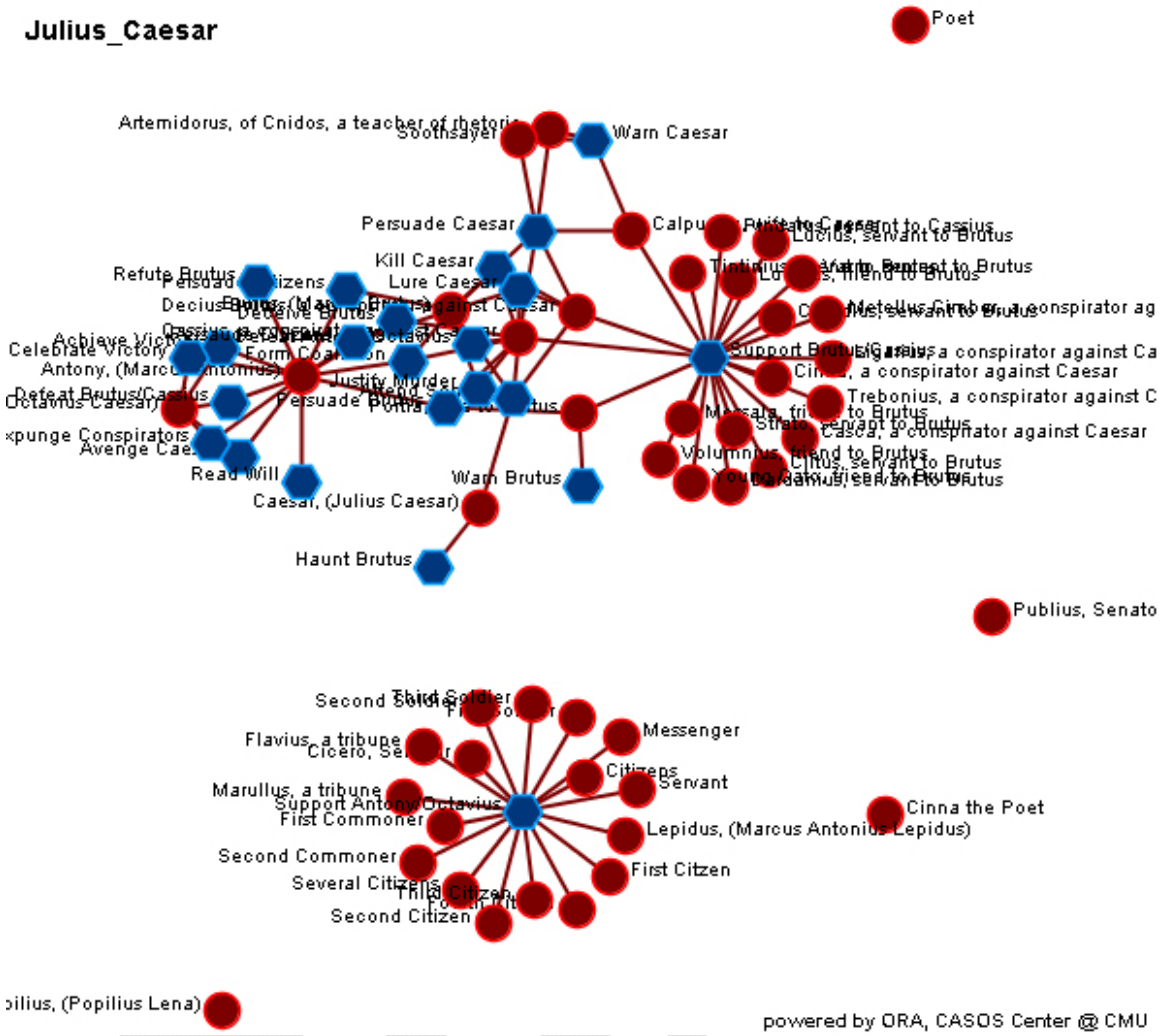


Agent by Task (Who is doing What)

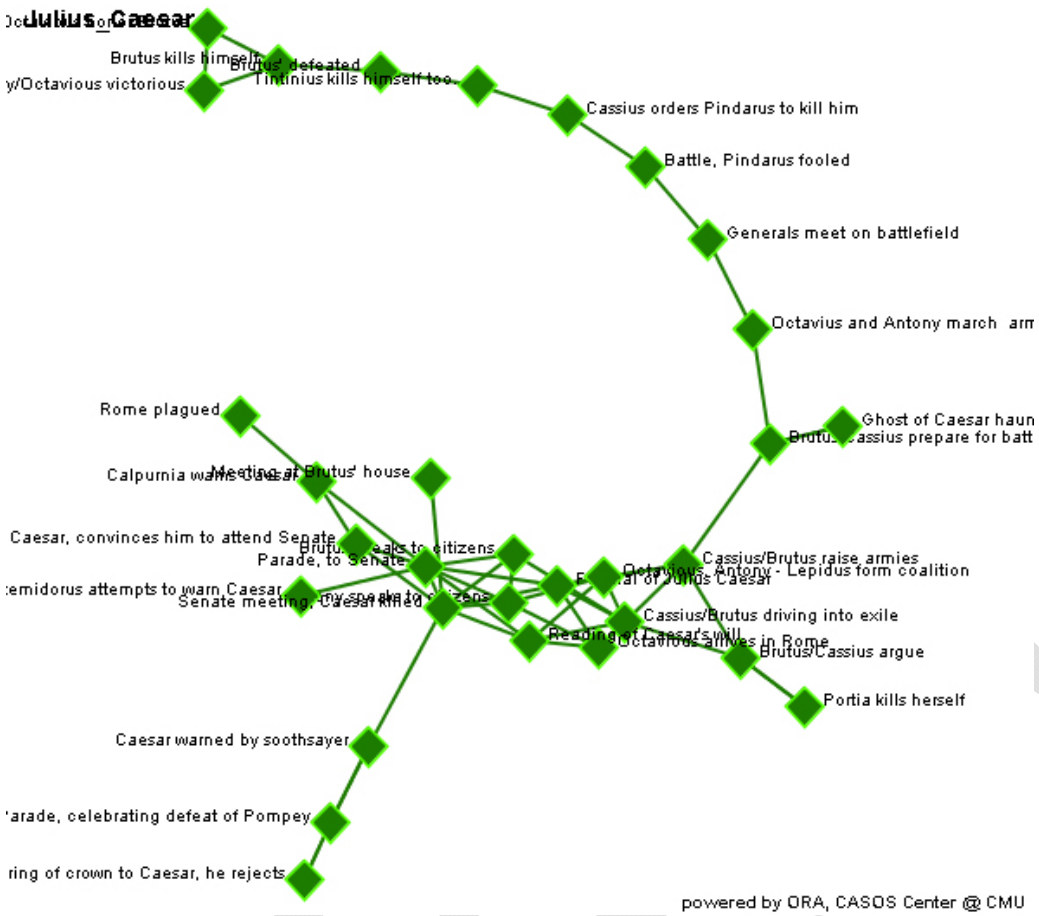


Task by Task (What needs done by What needs done)

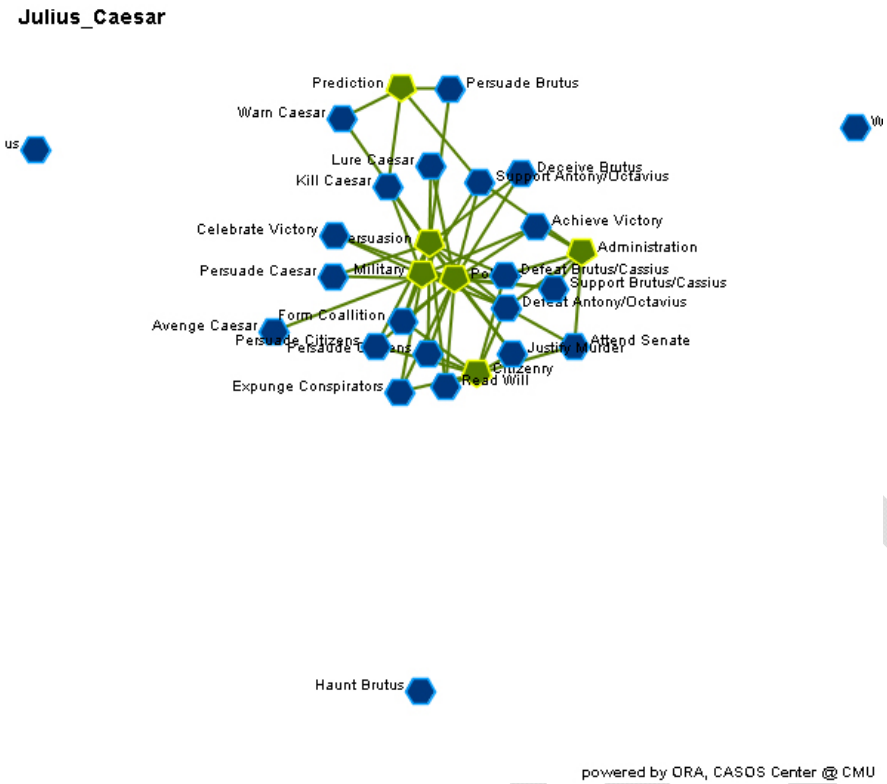

Julius_Caesar
Lure Caesar
Attend Senate
Persuade Caesar
Deceive Brutus
Kill Caesar
Persuade Brutus
Haunt Brutus
Justify Murder
Refute Brutus
Avenge Caesar
Persuade Citizens
Achieve Victory
Read Will
Celebrate Victory
Support Brutus/Cassius
Support Antony/Octavius
Defeat Antony/O
powered by ORA, CASOS Center @ CMU



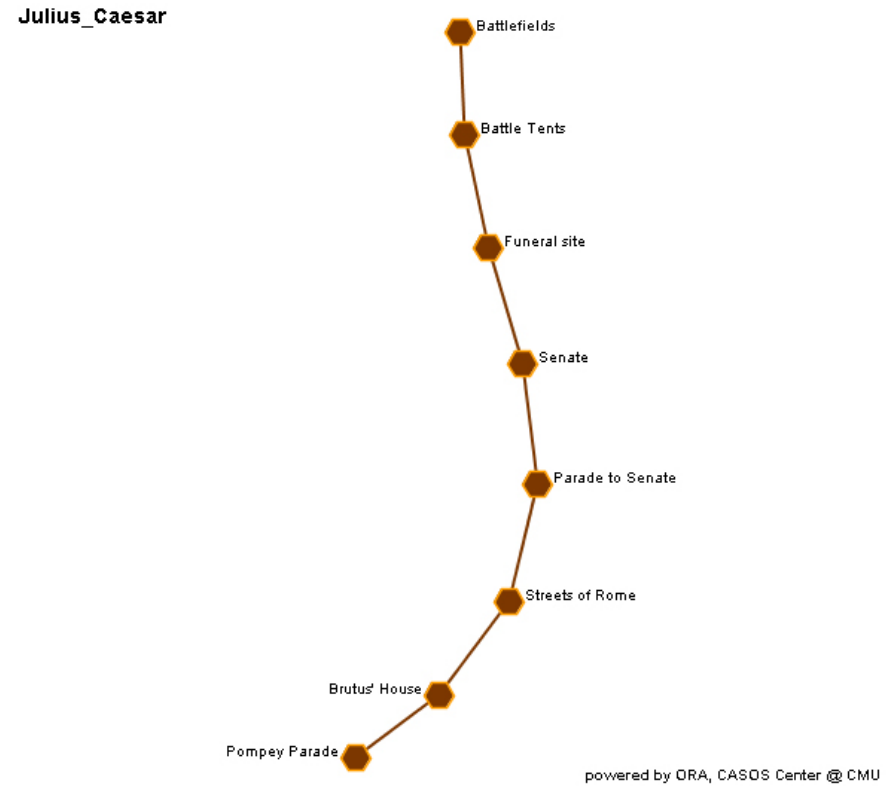
Event by Event



Knowledge by Task



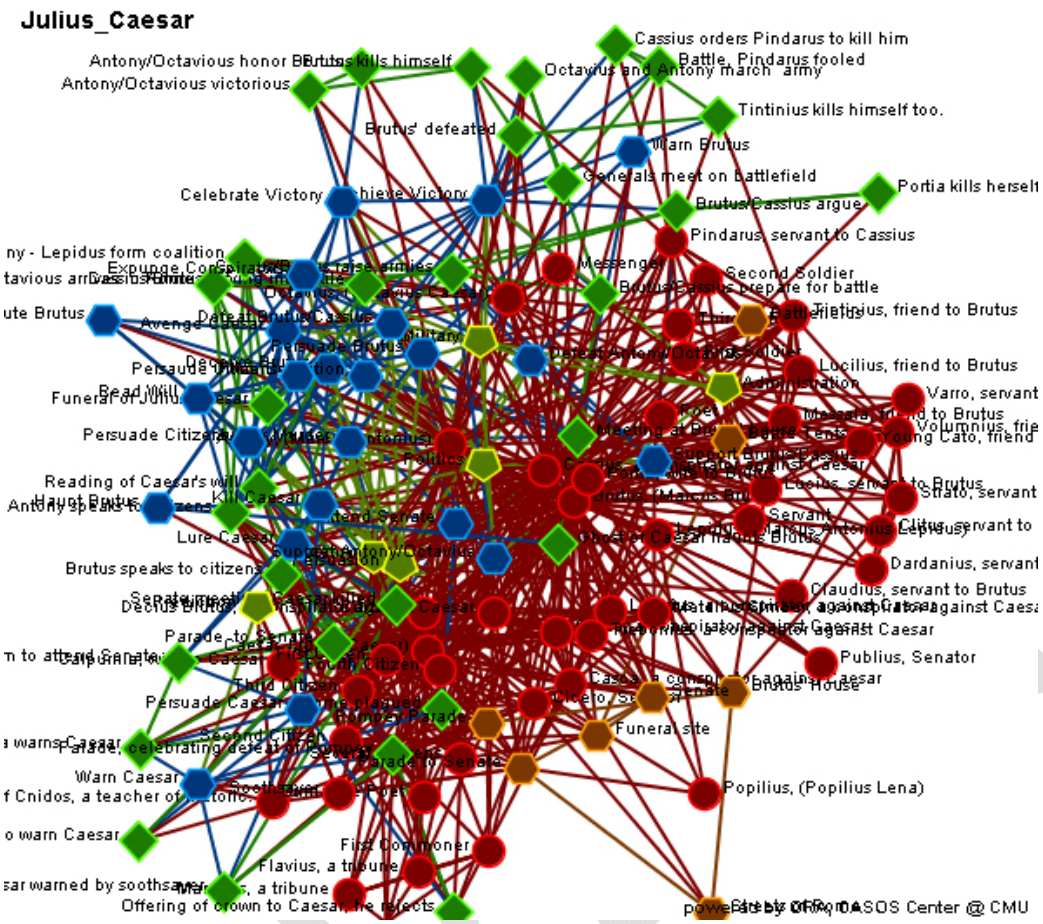
Location by Location (Where by Where)



Adding it all together – Our Julius Caesar MetaNetwork

So, now you have all these graphs of entities falling into many different entity classes. Now, using Dynamic Network Analysis we will add them all together – that is right – we said add them all together – to gain the most comprehensive, insightful and powerful model of a network that only DNA can deliver. The result is a graph based on graph Algebra – a summation of all the networks together. Below then is our final MetaNetwork of the Julius Caesar network and all inclusive graphs of Entity Classes.

The Julius Caesar MetaNetwork:



Now what?

Now you may be saying to yourself – just what kind of model are we now left with?

This complex MetaNetwork could be said to resemble a ball of yarn? What are we to make of it? Well, your reaction is to be expected because MetaNetwork analysis is beyond the intuition of the unaided mind and the resultant MetaNetwork can be hard for an analyst to comprehend. We also tend to call the MetaMatrix a Persian Carpet, because that is what it appears to look like with the myriad of ties going from one entity to another, like colorful strands in a Persian carpet. However, this is where powerful computational mathematics comes into play in the form of Measures and we can begin to break down the MetaNetwork and glean some powerful insight into the Networks architecture.

Chapter 3: Two Entity Classes

Julius Caesar wants to complete a thorough network annual analysis of how all of his senators, generals and administrators "connect" to each other so he can figure out if there are any within his own imperial ranks that would seek to depose him—information that would be valuable to any dictator right? Therefore, Caesar conducts a survey and soon discovers from the data *Who* is talking to *Who* inside his empire but he can't quite determine if by nature of these connections any of these pairs of communicators are a threat. After all, his model tells him very little about what they are talking about. His model tells him very about what tasks those that are talking share. His model tells him very little about what events those who are talking share in common. Moreover, it would be nice if Caesar had a model of how the knowledge of those who are talking is connected to events of those who are talking. Such a complex picture might indeed by what is needed to save Caesar from the Senatorial daggers (talk about a meeting gone bad – you may fret that your Power Point slide is out of order, but don't turn your back in the Senate chamber).

Clearly the complex information described above would prove extremely useful in providing an in-depth analysis of the communication channels relevant to Caesar's dictatorship right? It is the goal of chapter then to take this network data and draw useful conclusions about the nature of the network and how Julius might best optimize his operation and understand the dynamic relationships that are going on around him. Will this collection of network data allow Caesar to do just that?

Moreover, Julius Caesar is interested in a local rumored plot operating in the Roman imperial Senate. Caesar has identified several social networks containing a number of "persons of interest" who communicate with each other regularly by letter, chatting at parades, conversing in whispers, meeting in secrecy. Based on carefully obtained surveillance Caesar constructs a network model of who is talking to whom and come up with an elaborate map detailing these relationships. He carefully analyzes this data and draws conclusions about how best to disrupt this network. Once again, it is the hope of the network analyst inside Caesar that the data obtained reveals vulnerabilities within the network structure. The social network should provide that right?

Let us say Caesar is interested in which armories are accessible by which generals within his empire. He studies in close detail an inventory of the all the armories as well as the servers they are networked to and how they communicate with each other. In a sense he has established a *Who* is talking to *Who* network of the computers that run this institution of higher learning. He wants to take this information and draw certain conclusions about the way the network is structured so he can draw conclusions about how to make his military stronger. Will his current network model of how the armories are connected help him get the job done?

In all three examples we get social network data about Who is talking to Whom, be they worker conspiring senators in the Roman imperial senate, administrators and his trusted generals in the upper echelons of his government or how his armories are all interconnected. Caesar then proceeds to build a network model of all these entity classes as they are connected to other entities of the like kind. This is traditionally how link analysis worked, but is the best way to draw conclusions from traditional network data?

Is there better data available to Caesar at the tips of his fingers that might tell us something more useful than *Who* is talking to *Whom*? Is this information right in front of his very eyes? And if there are better questions to ask, do we have the data and types of data to get the answers that would prove most useful to Caesar? Additionally, once the data is collected, what can we do with it to get more out of it? In other words, how should we analyze this data if we were giving the task of helping Julius Caesar?

The answer to whether more valuable information can be extracted from our network models in the case of administrators and generals, the plot and the armory network is a resounding yes. We can do by

networking more than who knows Who, the DNA scientist would help Caesar build a model of Who knows What, Who is connected to What. In other words, there are indeed better questions to ask and yes we can get better answers that should help Caesar continue to manage his empire in the most efficient manner possible, ostensibly minimizing the chance he might be the victim of a ruthless plot to usurp him. The answer lies in building a network model of multiple entity classes.

In the first examples we clearly have an entity class of *Agent*, be it a person in a company, a terrorist in a cell or a computer in a computer network. In all three cases we have the senators talking to senators the agents and how they are connected to other agents; we have the senators and how they are connected to other senators in the organization and we have terrorists and how they communicate with each other. What happens if we add another entity class – such as resources? Would adding such an additional entity set expand and magnify what we might ascertain from our network data?

We would ask what resources do the agents, which are the senators, in our Julius Caesar plot, have in common. How are those resources networked amongst themselves in an Agent by Knowledge network? What greater level of complexity is suddenly revealed? What can the dynamic network analyst glean by doing so? How will this information put the dynamic network analyst into a position to offer even greater insight and analysis to Julius Caesar when it comes to shaping policy and an ensuing course of action that would help Caesar continue to manage his empire long into his golden years?

Why use such data

As you should be seeing by now, real networks are not one dimensional. People have knowledge, they have access to resources, events happen and change networks and certain nodal points will contain varying attributes distinguishing them from the others. In fact, *real networks* for that matter are rarely one dimensional except, perhaps, on the most abstract levels. Are those levels the ones we want to focus on when it comes to dealing with real networks and shaping policy? Probably not.

Meanwhile, most network analysts typically concern themselves with one entity class when they are studying networks. For example, Caesar is interested in how a network is formed amongst all the members of a particular Senate but fails to see the importance of how this network might be linked to other critical entity classes such as *resources*, knowledge or tasks for plotting and motivating those that could kill him. So, it would behoove Julius Caesar to learn how to connect his *Whos* to *Whats*.

It is the natural tendency when studying networks to make these sorts of analytical assumptions that we should be interested in how entities of the same type are connected versus an interest in more complicated network structure. It only seems natural. It is after all, the path of least resistance and that course is the one typically chosen by us given no knowledge about other paths available to us. However, by looking at how other entity classes related to other entity classes we began to discern more complex structure about how a network really functions and such network functionality is really what Caesar is after. Thus we began by asking what would be a more insightful relationship to model for Julius Caesar when it comes to our network data?

What other entities would become of prime interest to us in leveling an immense analysis of a social network and in understanding how it is interconnected to, say, a resource network or a knowledge network. Furthermore, how much greater worth would analysis be if key policy makers knew how affecting a resource network might translate to the disruption of a social network or conversely in making it more rigorous? And how might that network is expected to react over time? After all, nature abhors a vacuum and so does the dynamic network analyst.

Let us now resolve to consider the impact of interconnected entity classes such as knowledge and resources with our social network of agents and help Caesar out in his task of understanding the complex dynamic networks around him. So we will begin to model such networks using two entity classes, also called "bipartite data." To do so effectively, we need to be able to extract useful information from those

interconnected networks and make use of this bipartite data. We need to make careful observations for Caesar lest he not get the most out of his dynamic network model.

Two-Entity Class Measures

It is time to revisit "measures" once again. We learned in previous chapters that a measure tells us something unique about a network using computational mathematics. More specifically, a measure is an algorithm specially formulated to tell us meaningful information about network data that we apply our algorithm to. So far as the dynamic network analyst is concerned, measures can span two or even more entity classes. In this chapter we will be concerned with two entity classes.

Such measures are carefully constructed and arrived at by research and scientific method. These computational formulas are what drive the field of dynamic network analysis sand many of the mathematical measures represented breakthroughs on what it is possible in the area of dynamic network analyst. Once such tool developed to run these measures is the Organizational Risk Analyzer, which is constantly being updated and improved at the Center for Computational Analysis of Societal and Organization Systems at Carnegie Mellon University in Pittsburgh. Unfortunately, such tools would have been unavailable to Julius Caesar but we will humor ourselves and pretend that they are.

ORA is a computer based network analysis tool that detects risks or vulnerabilities of an organization's design structure. It can take as input multiple entity classes across a multiple networks (which we should know at this point are called MetaNetwork). As we have come to learn by now: the design structure of an organization is the relationship among its personnel, knowledge, resources, and tasks entities. These entities and relationships are represented by the Meta-Matrix or MetaNetwork. Measures that take as input a Meta-Network are used to analyze the structural properties of an organization for potential risk and the best way to do that is span the limitations of single entity input. We can add different entities to our MetaNetwork to arrive at for more useful insights about network structure. This should help Julius Caesar.

In fact, Caesar would be interested to know that ORA contain well over 100 measures which are categorized by which type of risk they detect. Measures are also organized by input requirements and by output. Right now we care going to examine certain measures that make the best use of two entity classes. We will soon learn there is much to discern from acquiring a MetaNetwork of certain entity classes. So hang on for the ride as we learn about the mathematics behind a select sample of these very powerful algorithms that reveal structure that social scientist and other disciplines have found of the most important to consider a network as a whole. We are confident Julius Caesar would be equally as interested. First, let us learn more about what relationships we hope to extrapolate from the Julius Caesar network by learning about a few more key DNA concepts.

Let us begin to look at bipartite data as it would be visualized in our Two-Entity Class model of the Julius Caesar Network.

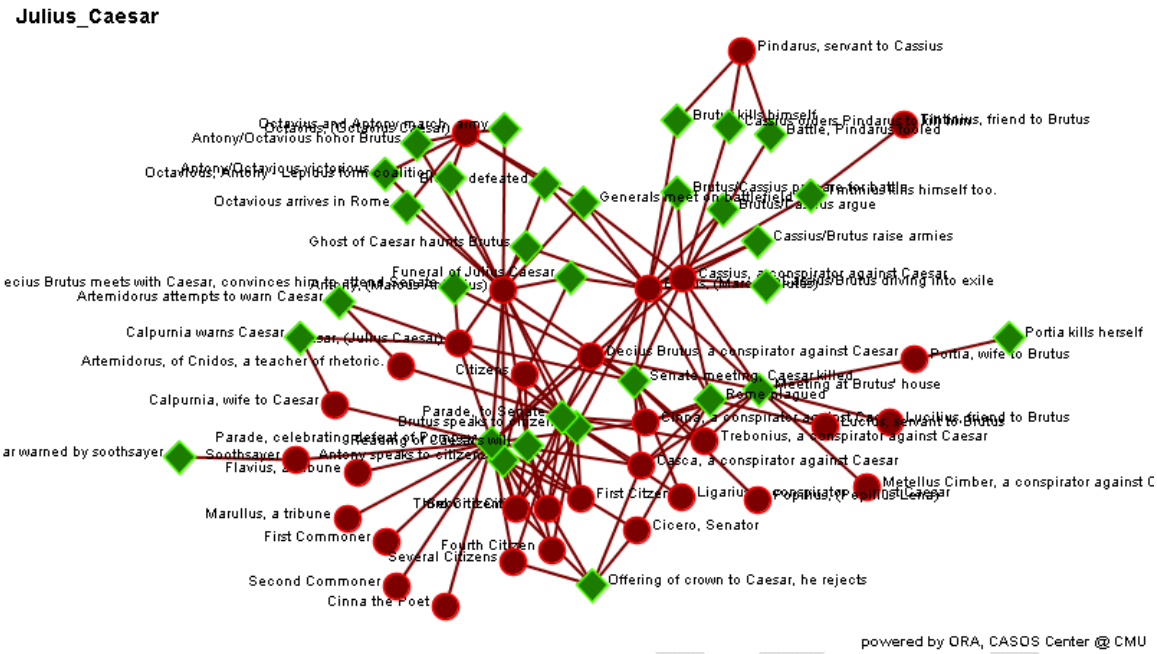


Figure 16: Agent by Event

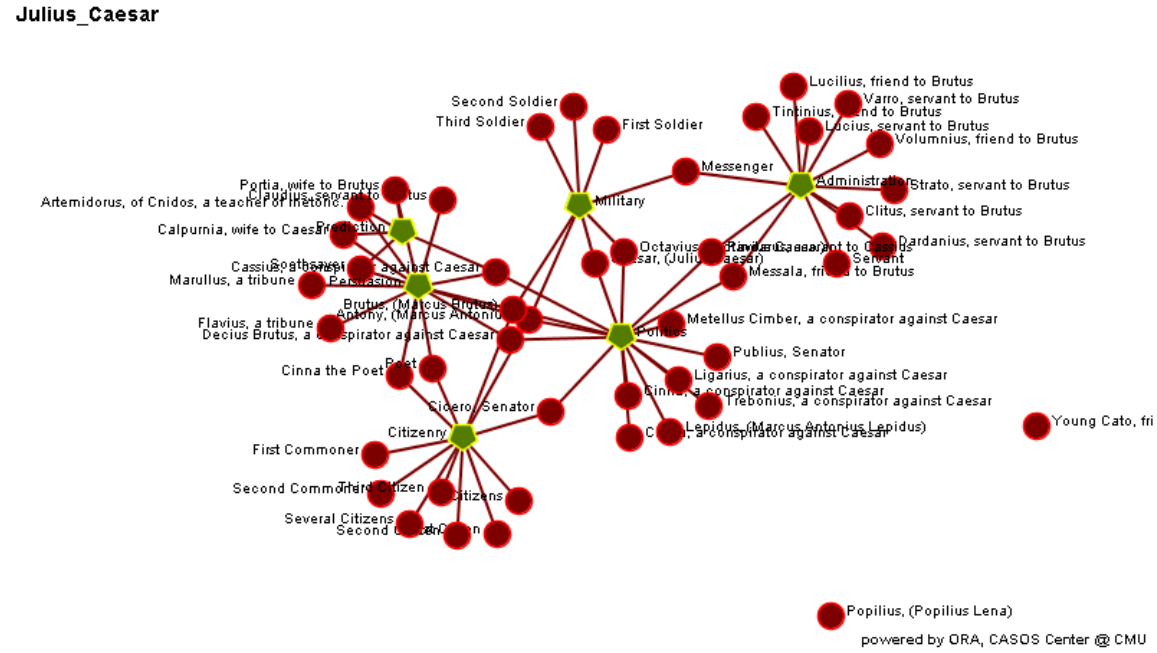


Figure 17: Agent by Knowledge

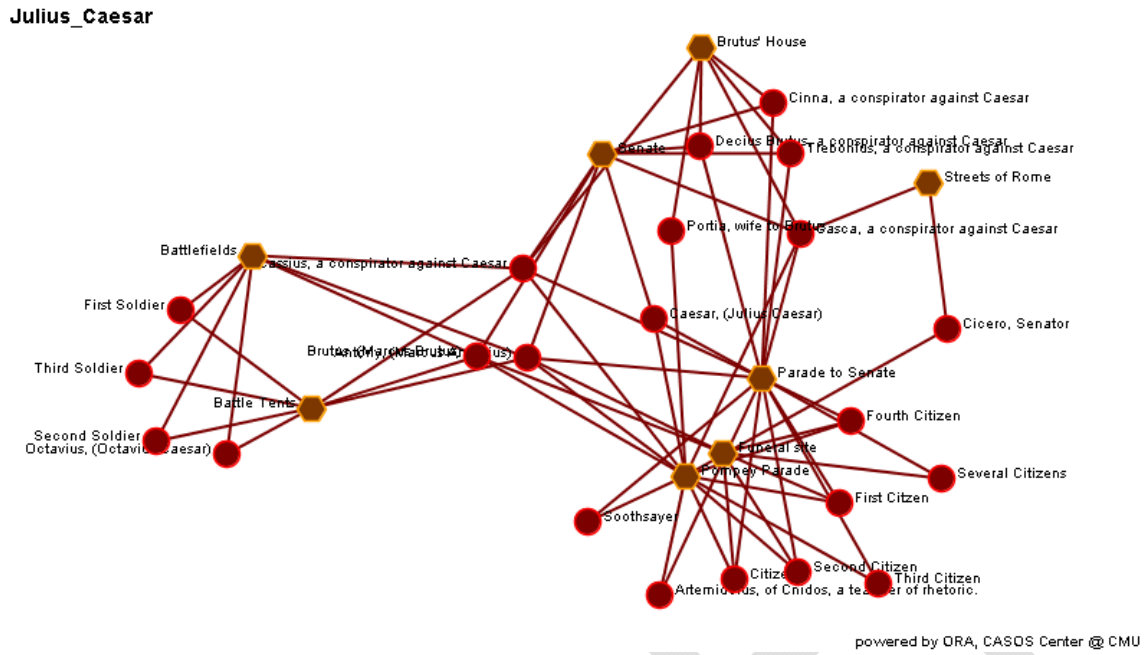


Figure 18: Agent by Location

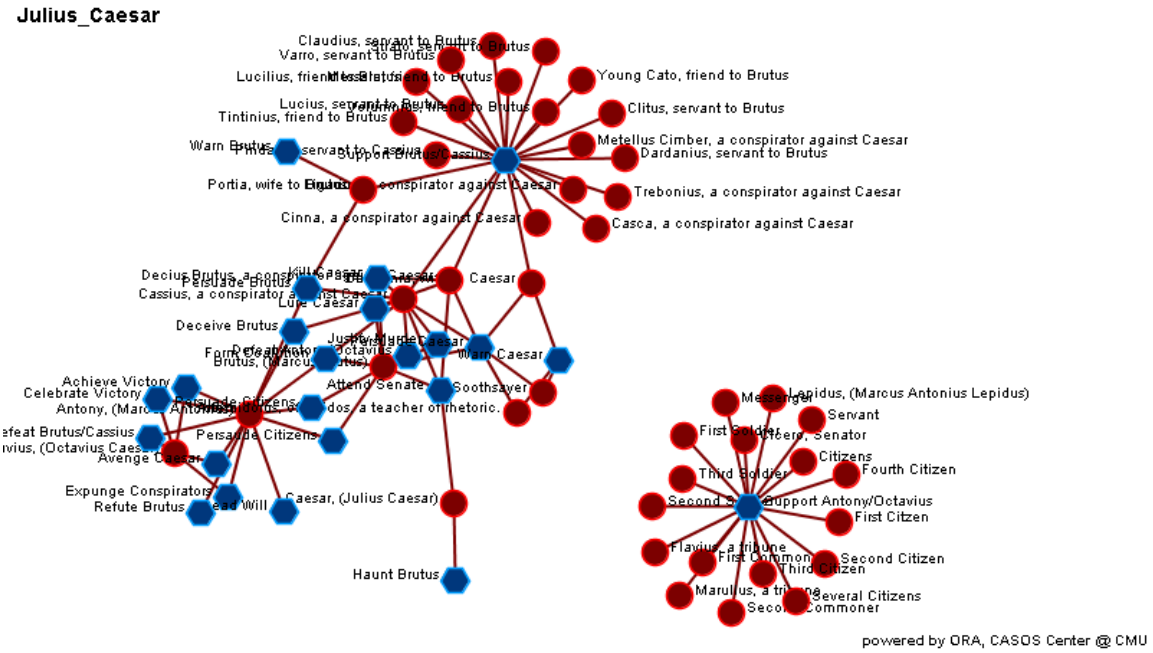


Figure 19: Agent by Task

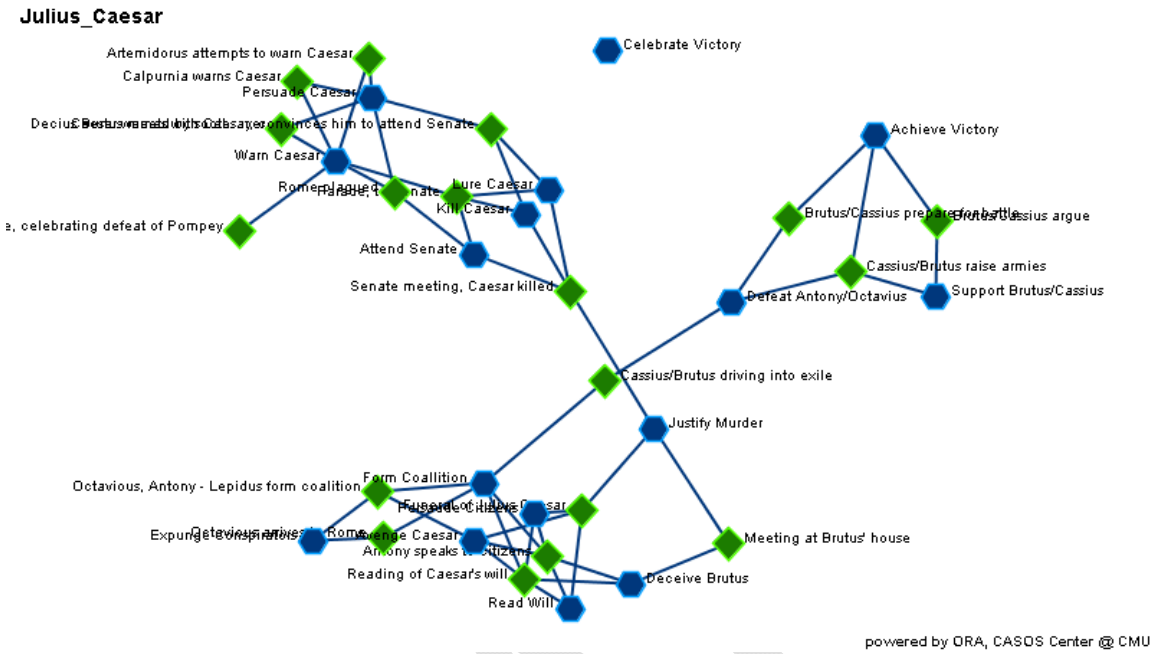


Figure 20: Knowledge by Task

Exclusivities

Exclusivity in network analysis can be of fundamental interest to discerning the critical infrastructure of a network. Naturally, as the term may allude, we are interested in performing an analysis that will reveal exclusive entities within our network model. Now, you may first ask what exactly do we mean by exclusivity? Generally, speaking we are looking for any entity within our network model that has an exclusive tie to another entity, which occurs across our bipartite MetaNetwork data. In lay parlance, this is finding out who in the network does something that nobody else does.

If we are helping Julius Cesar, the Dynamic Network Analyst decides to perform an exclusivity network analysis to determine Who in the company does something that nobody else does, or has access to a resource that nobody else does, or maybe has knowledge that nobody else does. Such information could be very useful to Caesar in determining his empires, eh, organizational vulnerabilities. After all, if "Cassius" is the only one that knows how to motivate Brutus to motivate the other senators, what happens if Cassius decided to turn against him as we know he eventually does?

Revisiting our previous examples, we have to ask ourselves how important exclusivity is in the analysis of a senatorial plot to dethrone Caesar by assassination. If one particular agent in the plot is the only one that can motivate Brutus to led a rebellion, yet is pretty low ranking in terms of the hierarchy in an Agent by Agent network, how useful would that information be to Julius Caesar? There you have a key vulnerability that could foil the entire assassination plot right at your very own dagger tip.

Now we have our network of generals and military advisors to Julius Ceasr. We have over 50 computers networking with say 5 generals. However, say 1 of the 5 generals performs a critical function that if he went down, the other 4 generals could not replicate the role of the first. What a vulnerability that would be. Our exclusivity measure could look at the is network of resources and agents and determine where such exclusivity may rest. How valuable would such information be to the network analyst interested in understanding how to strengthen the Julius Caesar network?

The math behind Knowledge Exclusivity

The Knowledge Exclusivity Index (KEI) for agent i is defined as follows:

Resource Exclusivity Index (REI) or Task Exclusivity Index (TEI)

$$\sum_{j=1}^{|K|} AK(i, j) * \exp(1 - \text{sum}(AK(:, j)))$$

*For **Resource** replace AK with AR*

*For **Tasks** replace AK with AT*

*For **Knowledge** replace AK with AT*

The Resource Exclusivity Index (REI) for agent i is defined exactly as for Knowledge Based Exclusivity, but with the matrix AK replace by AR.

The Task Exclusivity Index (TEI) for agent i is defined exactly as for Knowledge Based Exclusivity, but with the matrix AK replaced by AT.

In the formula above A is representing an "Agent" entity where is K equates to "Knowledge." By exchanging the values for Knowledge with Tasks and Resources we will arrive at entities with those respective exclusivities. This demonstrates the added advantage of using bipartite data: it adds another entire dimension to the science of network analysis, thus making it dynamic.

Specialization

In some sense, Exclusivity is highly related to the issue of specialization. It is critical to the dynamic network analyst to understand Who specializes in what within any network. Agent by Agent graphs simply will not do the trick to ascertain specialization. However, using bipartite data we can begin to extrapolate such specialized roles within our network by running our bipartite based measures on our MetaNetwork data. The value of this should be obvious.

Within the Julius Caesar network it would be highly valuable to understand Who amongst all of his underlings possesses a specialist knowledge relating to the operation of his empire. Maybe the only way to get this information is to look at the bipartite data set of Knowledge x Agent or Task x Agent or Resource x Agent. Maybe a certain amount of unique ties to agents and tasks constitutes a specialist. It is such unique relationships we are trying to discover by employing this sort of data.

What exactly do we mean by specialization as it applies to networks? We are interested in the adaptive nature of certain entities to a particular task.

Let us say within the Julius Caesar network we have a certain agent with certain *knowledge*. Perhaps this information will tell us something about how this agent, this node, this person will adapt to the network over time, that is become specialized.

In a terror network it should be inherently clear as to the advantages of discerning the adaptive nature of any given entity that is an agent, in the terror network.

If we network data on one terrorist, perhaps then we ascertain what his tasks are now and how he will adapt given a period of time. We might be able to come up with the odds that this particular plotter will eventually carry out this particular task. He will be part of network of specialization.

It is making use of bipartite data that we can run highly tailored measures used to provide this crucial information. But, using bipartite data allows us to glean other structure that is highly valuable to the network analyst.

Redundancies

Redundancy more formally acknowledged by the dynamic network analyst is the risk based on duplication in task assignments, resource access, and knowledge access. An organization with little redundancy is more adversely affected by an agent or resource no longer being available such as a "Cassius" taking his knowledge and getting a new job in some other empire. On the other hand, too much redundancy makes an organization inefficient. For instance, you wouldn't want everybody in the imperial Senate to learn how to conduct military operations when everyone has other tasks and accountabilities to see too as well.

Likewise, in a terror network, it would be greatly insightful to learn which agents within the network are redundant. If everybody has the "knowledge" bomb making, then what would be a more effective strategy than dealing with this network? We would clearly see that the *knowledge* of bomb making is redundant through the network and therefore eliminated any agent at random would likely result in a minimal disruption to the network. However, if ascertained that they all know how to make bombs, but only one terrorist in particular has access to the materials, that is he has exclusive knowledge in knowing where to locate the materials, then suddenly a better strategy emerges. In fact using non bipartite data, exclusivity and redundancy are largely negated save for the top level connections agents might share with each other.

It is a more complex picture when we use bipartite data and calculating agent x knowledge, task and resource reveals the strengths and vulnerabilities of the network.

The math behind Redundancy

Redundancy, Access

Average number of redundant agents per resource. An agent is redundant if there is already an agent that has access to the resource.

Carley, 2002

TYPE: Graph Level

INPUT: AR:binary

OUTPUT: $\Re \in [0,(|A|-1)*|R|]$

This is the Column Redundancy of matrix AR.

Redundancy, Assignment

Average number of redundant agents assigned to tasks. An agent is redundant if there is already an agent assigned to the task.

Carley, 2002

TYPE: Graph Level

INPUT: AT

OUTPUT: $\Re \in [0,(|A|-1)*T]$

This is the Column Redundancy of matrix AT.

Redundancy, Column

The mean number of column node edges in excess of one.

TYPE: Graph Level

INPUT: N of dimension m x n

OUTPUT: $\Re \in [0,(m-1)*n]$

Let M be the matrix representation for a network N of dimension m x n.

let

$$d_j = \max\{0, \text{sum}(M(:, j)) - 1\}$$

for

$$1 \le j \le n$$

this is the number of column entries in excess of one for column j.

Then

$$\text{Column Redundancy} = \left(\sum_{j=1}^{n} d_j \right) / n$$

Redundancy, Knowledge

Average number of redundant agents per knowledge. An agent is redundant if there is already an agent that has the knowledge.

Carley, 2002

TYPE: Graph Level

INPUT: AK: Binary

OUTPUT: $\Re \in [0, (|A| - 1) * |K|]$

This is the Column Redundancy of matrix AK.

Redundancy, Resource

Average number of redundant resources assigned to tasks. A resource is redundant if there is already a resource assigned to the task.

Carley, 2011

TYPE: Graph Level

INPUT: RT:binary

OUTPUT: $\Re \in [0, (|R| - 1) * |T|]$

This is the Column Redundancy of matrix RT.

Redundancy, Row

The mean number of row entity edges in excess of one.

TYPE: Graph Level

INPUT: N of dimension m x n

OUTPUT: $\Re \in [0,(n-1)*m]$ for N dimension m x n

Let M be the matrix representation for a network N of dimension m x n.

let

$$d_i = \max\{0, \text{sum}(M(j,:)) - 1\}$$

for

$$1 \le i \le m;$$

this is the number of column entries in excess of one for row i.

Then

$$\text{Row Redundancy} = \left(\sum_{j=1}^{m} d_j \right) / m$$

Inferring relations

The Julius Caesar network is quite large at 48 but it is not quite so large that we cannot capture all of the network information ourselves on an agent by agent basis. But that is not really that large, compared to say a modern company of over 100,000. When you are talking in such large numbers like this, things change a bit. In fact, it is too large to really capture all the network data on the entire company. We simply don't have the time to talk to each and every agent to build a network model. After all who has the time? The same would apply to Julius Caesar's empire. Say his empire grew to over a 100,000 employees spread out across 5 continents. How long would it take to gather all the necessary data to perform an analysis on the entire network? Perhaps it would not be impossible, but would it be practical?

You have to consider that real networks are rarely small enough that we all the data. Even in a network of 48 agents such as in the Julius Caesar play, how much more complex do our data requirements become when we add resources, knowledge, tasks and events that affect the organization? Picture how this would multiply out with a 100,000 employees, should suddenly the Julius Caesar Empire expands? Does that mean we are hamstrung by our efforts because we can only perform dynamic network analysis on small real networks? Not hardly.

We go back to our measures, the heart of dynamic network analysis.

The measures developed at CASOS in particular are designed with the fact that we usually will have to infer relationships, that is ties, between entities in real networks. There are simply too many relationships, too much data to consider. Not enough time to do a survey of the entire network. We have to go on partial information. In that sense, dynamic network analysis is similar to quantum physics,

nodes and relationships are "probabilistic." It is much like the scientific application of the statistical analysis. It is impossible to conduct a poll on each individual on most topics, but if done scientifically, then it can be done with high levels of confidence that what you are polling is what the populace as a whole believes. The same would apply to dynamic network analysis and the same would surely apply to Julius Cesar's empire.

The work-around on this problem is that our measures are designed to tell us probabilities that let us know the likelihood that any given entity in our MetaNetwork will likely have a tie, or relationship, to a certain resource, knowledge or task. Once again, we are underlining the value of using bipartite data to perform network analysis. If we know a certain agent has a particular knowledge, we might be able to quantify the probability he might have a relationship with other entities.

For instance, let us say we partial network data on one of Julius Caesar's armies. We have one agent; w will call this entity "First Soldier". We know that First Soldier is a is encountered quite often in many acts with Second Soldier and Third Soldier. However, we don't know exactly what resources he has access too or what knowledge he actually possesses but we do know that information about Second Soldier and Third Soldier. Now let us say Julius Caesar wants to know what connections First Soldier is likely to have within his empire.

Traditional link analysis is rendered moot. We simply would not have an answer because First Soldier never took the employee survey sent out a month ago. The Dynamic Network Analyst has complete data on Second and Third Soldier, who do exactly what Bob does. Would it be reasonable to suggest First Soldier had the same skills as Second and Third Soldier? Perhaps. If it were deemed that we had a thousand more soldiers, maybe we would feel even more confident they would all explain the sort of characteristics that First Solder would likely have.

The Dynamic Network Analyst can look at this data and apply exclusivity or redundancy measures or other measures relating to specialization and get a profile of what our ties our average entity is likely to have. Once again, we underpin these inferences on the power of our algorithms in our measures. If any soldier in our sample contains an average education of a military school, has an average of military ties, and has access to X amount of military resources, then painting a picture of First Soldier just might be within the realm of the analyst – the dynamic network analyst that is.

Chapter Summary

By now we should see the inherent benefits of using bipartite data, which is network data that in the form of a MetaNetwork spans two entity classes. We can see that by moving beyond MetaNetwork analysis of one entity class, that is agent by agent, we can learn much more complex and valuable information about a real a network structure. This is structure that is unobtainable in traditional one-entity evaluation of a network. This is information that could prove highly valuable to Julius Caesar

Employing bipartite data we learned that we can apply powerful computational mathematical formulas to help us discern this structure. Such algorithmic formulas are measures and they are constantly being developed at CASOS for the use of the Dynamic Network analyst.

We learned about exposing the strengths and vulnerabilities of a networks structure by revealing the redundancies of any entities that comprise the network and the exclusive tasks contained therein by those respective entities. We can see that this is made possible from the added dimension of using bipartite data.

We also touched upon the adaptive nature of any entities within our network and about inferring relationships on partial data. Often time complete information on a real network is a pipe dream and we have to infer relationships based on the data samples we are able to obtain and the Julius Caesar model is no exception. Using bipartite data we can infer quite a bit more than if we otherwise relied solely on one entity class of information.

In the next chapter we are going to explore measures that go beyond even bipartite graphs or data. After all, if we take a MetaNetwork made up of agents, knowledge, events, tasks, locations, there must be even greater complexity at our disposal. Such data adds a third entity class. In the next chapter then we will explore measures using three entity classes. After all, we have to provide Julius Caesar with the best network analysis possible. To do so otherwise, might not just cost him his life but ours too, perhaps, if he is not happy with our work. We don't' want that to happen.

Chapter 4: Three Entity Classes

Julius Caesar evaluates his network data of his current political advisors. He takes a good look at the personal in the empire and what their role is in keeping his empire functioning. One of the political apparatchiks is instructed to create a new midlevel advisory position to report directly on the doings of the rank and file in Caesar's empire. For the sake of this example, we will say that the person given the unenviable task is Marc Antony, who is not sure how best to go about finding the best candidate for the new job or who would be the best candidate based on the survey of information they obtained about the rank and file. What should Marc Antony do? Who can he turn to? What would be his strategy to cull the best person out of his field of applicants?

Should Antony post a scroll with a minimum set of qualifications, and then seek applications from the rank and file. Typically those applications would come in the form of resumes, at least today, but in our case, we are sure what would suffice to serve as a resume. The midlevel manager at ABC would then quickly scan the applications attempting to narrow down the field for the best possible candidates. How does the typical midlevel manger go about doing this? One would think he would take the time to carefully scrutinize and peruse every application that came across the midlevel manager's desk, but you and I both now that usually doesn't happen.

Sometimes, as is it is far too often the case, the best candidate is whittled out from the application process due to arbitrary requirements, many times designed to speed up the process of finding the best applicants but oftentimes resulting in the unintended consequence of eliminating the candidate with the greatest potential. However, what if merely the misspelling of some term got such and such application remediated to the unqualified pile.

Let us say the first applicant simply was adept at making sure the I, III and Vs were properly displayed in his Roman documents. This person would know the key words to press with the hiring manager but would not necessarily be qualified to do the job at hand. He was surely accountable and competent at his job, but was he the real rising star in the political ranks? Ultimately, you can make the case that his attention to detail is lacking. Perhaps this alone should be reason enough to not hire the applicant. However, can you really say that such a transgression should eliminate an applicant who had the highest measurable value of knowledge, access to resources, and the other skills necessary to be successful as Caesar's next advisor? At the very least, if one had this information about the applicant, one could give him the benefit of the doubt.

Now, keep in mind, we are not talking about blatant and willful disregard for standards and protocols, but a small typo, perhaps not even the fault of the applicant. Should such a small error could keep you from meeting with this person, who just might be the best person to assume Caesar's next potential political advisor role.

Marc Antony would need a tool to calculate who is highest among his resources, knowledge and connections. Would you want to talk to the people at the top of the list? Caesar surely would.

Let us consider the task of analyzing Caesar's network as a global operation. Here we have this enormous network model spanning the Mediterranean. Julius Caesar is interested now in identifying the most important people in the organizational structure that could dramatically impact the network structure if such individuals were "removed" from the network. However, who would be such a person? Naturally, we might say Brutus, but as of now, we don't know where he is. Let us say that Caesar however has closely monitored a cell of individuals operating in the Senate. This cell is made up of over a handful of individuals, of which Cassius and Brutus are the most prominent. Each of them shares unique duties relating to the cells operation. It is not clear which ones within the cell or the most integral to supporting the cells structure and the continuance of its operations. Could it be that one of the men with the least amount of responsibilities could evolve to be the next Caesar, if circumstances permitted? Would it be useful to have a method to determine who that next Caesar might be based on analyzing our network mode? Of course it would. Caesar would agree.

Let us now think more in terms of the overall "big picture" we have of the cell. Would it be useful for the military analyst in Caesar's court to know if certain agents within a given group of senators had the right tools to usurp his reign? That is these senators had the best resources, needed to carry out the goal of the organization? That is getting into a rather complex insight into network structure but one that could be deemed highly insightful to Caesar's military analysts. What if we were interested in knowing the skill sets certain agents lack in carrying out their tasks? Is that at our finger tips by carefully analyzing our model? Would it be useful to look at the model and then come up with an index number of how efficient the cell is at carrying out its tasks with other cells that have the same tasks? If so, how would we go about doing this?

Perhaps now we see the value of using bipartite data to tie in several network entities classes into the mix such as resources and knowledge. What our Meta-Network now gives to us is more than two classes, but three. Now we have a Meta-Network model of agents, resources and knowledge as it pertains to the senatorial network? What useful information could Caesar's military analyst now make use of in this scenario?

Let us revisit our administrative advisor from the previous chapter. This is the one with the enviable task of making sure the bureaucracy of Caesar's empire is optimized in terms of allocation and other resources. Does our advisor understand how to look at Caesar's empire network and ascertain, if based on the rank and file, Caesar is indeed allocating his network resources in the most efficient manner? Maybe reality holds that the most powerful administrators are essentially going untapped as they are asked to do tasks that less expensive apparatchiks could do at a fraction of the costs. We can presume Caesar would surely like to know this information as well as would the top levels of any large organization.

If the administrative advisor had all this information in his network model of agents could our advisor be able to develop a plan to best reconfigure the Empire so that those administrators with the most demands have access to the resources most capable of delivering the solutions needed by those administrators?

The administrator needs to understand what his network model can tell him about allocating such resources. He should by now be familiar with the inherent benefits of the using two entity class data, bipartite, to calculate certain features of his network, but now it becomes a little more complex. He is not merely interested in an individual network, but rather the entire network as a whole.

Maybe now he is given data on another part of the Empire that is considering the optimum balance in military resources and political demands. If he wanted to straighten up compare his political network with that of the ideal computer network on what basis could those two networks be judged? Caesar would be most interested.

Now that we are familiar with the joys of bipartite graph data and can see the application using such data has to the dynamic network analyst, we are going to add a third piece of data to the mix. Thus we are going to consider three entity classes.

Why use such data

As Julius Caesar learned in the previous chapter, real networks are not one dimensional. People in his empire have knowledge, they have access to resources, events happen that can fundamentally alter networks and certain nodal points will contain varying attributes distinguishing them from the others but there is even more to dynamic network analysis than just that. We know *real networks* for that matter are rarely one dimensional except, perhaps, on the most abstract levels and we know that those levels generally don't tell us exactly what we want to focus on when it comes to dealing with real networks and shaping policy

As we learned, most network analysts typically concern themselves with one entity class when they are studying networks. In the previous chapter we started considering two entity classes. For example, we were interested in how a network is formed amongst all the members of a particular company or how a group of senators in Caesar's empire are all connected but failed to see the importance of how this network might be linked to other critical entity classes such as *resources* for supplying the military, perhaps money to finance its operations, and *knowledge* development and training. We learned we could begin to see much more complex structure by using bipartite data.

We learned about which nodes in the network were unique, that is specialized, which ones were redundant, pretty much useless and which ones had properties we were able to infer to other nodes in the network. This was all of interest to Julius Caesar.

We also talked about the natural tendency when studying networks to make these sorts of analytical assumptions that we should be interested in how entities of the same type are connected versus an interest in more complicated network structure. It only seems natural. It is after all, the path of least resistance and that is the course with the one typically chosen by us given no knowledge about other paths available to us. However, by looking at how other entity classes related to other entity classes we began to discern more complex structure about how a network really functions.

Thus we began by asking what would be a more insightful relationship to model when it comes to our network data. Once again, we are doing the same process, but only adding even more complexity.

What other entities would become of prime interest to us in leveling an immense analysis of a social network and in understanding how it is interconnected to, say, a resource network or *knowledge* network in Caesar's empire? Furthermore, how much greater worth would analysis be if key policy makers knew how affecting a resource network might translate to the disruption of a social network or conversely in making it more rigorous? And how might that network is expected to react over time?

Let us now consider the impact of three interconnected entity classes such as knowledge and resources and tasks as they apply to our previously mentioned social network of agents. Again, just as we learned in the previous chapter it is time to consider measures.

This time our measures will incorporate three entity classes to reveal structure that would allow us to answer all of the aforementioned questions previously outlined.

Cognitive Demand

Cognitive demand is a measure that tells the dynamic network analyst quite a bit about the unique properties of an agent in relation to the whole network in which they are integral part. Aan agent high in the cognitive demand value would be such an agent a network analyst might term a "leader" or even an "emergent leader." Caesar surely would be interested in learning about who his emerging leaders are?

In the case of our CIA operative, knowing the cognitive demand value of an agent can be invaluable. It could tell the analysts that such and such a terrorist might figure prominently in the terror cell as an up and comer, someone who might need to be reckoned with in the near future. The final analysis could be that is better to get him now, rather than later. Moreover, identifying agents that are high in cognitive demand is like knowing which agents in a network that if removed could substantially undermine the structure of the network as a whole.

In these measures, we are ascertaining complex structure by considering a third entity class. We are now employing a Meta-Network of agents, knowledge and tasks. The most essential to calculating these complex measures is having an agent by agent graph. Whatever other entity classes we wish to add to the mix, are optional so to speak. However, using different combinations therefore of three entity classes reveals the measure of cognitive demand.

Back to Cesar's network they could use the measure for cognitive demand and see who might really be doing the hard work in keeping Caesar's Empire functioning. In such as network model, based on connections to resources and tasks, we would determine the highest cognitive demand. This would essentially present the analyst a good picture of an up and comer at ABC Corporation. Maybe then, they should give the job to this applicant or at let Caesar know he has a real star on his hands.

In our resource by resource model, we might learn which skills are carrying the most user load and performing admirably in that regards. There might be a good servant to upgrade that is worth promoting to a more prominent part of the overall network.

It is clear to see why Cognitive Demand and the variations of this measure are so important. Once again, we see that such measures are based on the framework of the Meta-Network, which spans more than one entity class.

The Math behind Cognitive Demand

Cognitive Demand

Measures the total amount of effort expended by each agent to do its tasks. Measures the total cognitive effort expended by an agent to do its tasks.

Individuals who are high in cognitive demand are emergent leaders. Removal of these individuals is quite disruptive to networks.

Note: The minimum input requirement is the AA network. All other networks are optional.

Carley, 2002

TYPE: Agent Level

INPUT: AA:binary; [AT:binary]; [AR:binary]; [RT:binary]; [AK:binary]; [KT:binary]; [TT:binary]

OUTPUT: Agent Level:

$$\Re \in [0,1]$$

The Cognitive Demand for an agent i is an average of terms, each of which measures an aspect of its cognitive demand. Each term is normalized to be in [0,1]. The number of terms depends on the input networks. The computation of each term for agent i is detailed below:

let x1 = # of agents with which i interacts

= sum(AA(i,:)) / (|A|-1)

let x2 = # of tasks to which i is assigned

= sum(AT(i,:)) / |T|

let x3 = sum of the # of agents assigned to the same tasks as i

= sum(ATA(i,:)-ATA(i,i)) / (|T|(|A|-1)), where ATA = AT*AT'

let x4 = # of resources i manages

= sum(AR(i,:)) / |R|

let x5 = # of knowledge i manages

= sum(AK(i,:)) / |K|

let x6 = sum of # resources i needs for all its tasks

= sum(ATR(i,:)) / (|T|*|R|), where ATR = AT*RT'

let x7 = sum of # knowledge i needs for all its tasks

= sum(ATK(i,:)) / (|T|*|K|), where ATK = AT*KT'

let x8 = sum of resource negotiation needs i has for its tasks

= HammingDistance(AR(i,:), [AT*RT'](i,:)) / |R|

let x9 = sum of knowledge negotiation needs i has for its tasks

= HammingDistance(AK(i,:), [AT*KT'](i,:)) / |K|

let x10 = sum of agents that i depends on or that depend on i

let w = # number of agents assigned to each task

= colsum(AT)

let s = # agents that dependent on each task

= (T+T')*w

let v = # tasks that agents are dependent on

= AT*s

Then, x10 = v(i) / (|A|*|T|*(|T|-1)).

Then Cognitive Demand for agent i is the average of the above terms.

Cognitive Resemblance / Cognitive Resemblance, Relative

Measures the degree of resemblance between agents based on the number of knowledge bits they both have or both do not have.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT: $\Re \in [0,1]$

For each pair of agents (i,j) compute the number of knowledge bits they have in common - whether known or unknown. Then normalize this sum by the total number of knowledge bits.

$$CR_{i,j} = \frac{\sum_{k=1}^{|K|}(AK_{i,k} * AK_{j,k}) + (\sim AK_{i,k} * \sim AK_{j,k})}{|K|}$$

$$CR_{i,i} = 1$$

Note that the CR output matrix is symmetric.

Relative Cognitive Resemblance normalizes each element of CR as follows:

$$RCR_{i,j} = \frac{CR_{i,j}}{\sum_{j=1}^{|K|} CR_{i,j}}$$

Thus, the elements of the ith row are normalized by the ith row sum.

Cognitive Similarity / Cognitive Similarity, Relative

Measures the degree of similarity between agents based on the number of knowledge bits they both have.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT: $\Re \in [0,1]$

For each pair of agents (i,j) compute the number of knowledge bits they have in common. Then normalize this sum by the total knowledge between them.

$$CS_{i,j} = \frac{\sum_{k=1}^{|K|}(AK_{i,k} * AK_{j,k})}{\sum_{k=1}^{|K|}(AK_{i,k} + AK_{j,k})}$$

$$CS_{i,i} = 1$$

Note that the CS output matrix is symmetric.

Relative Cognitive Similarity normalizes each element of CS as follows:

$$RCS_{i,j} = \frac{CS_{i,j}}{\sum_{j=1}^{|K|} CS_{i,j}}$$

Thus, the elements of the *ith* row are normalized by the *i*th row sum.

Cognitive Distinctiveness / Cognitive Distinctiveness, Relative

Measures how distinct are two agents based on the number of knowledge bits they hold oppositely.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT: $\Re \in [0,1]$

For each pair of agents (i,j) compute the number of knowledge bits they have exactly opposite. Then normalize this sum by the total number of knowledge bits. In effect, this is the exclusive-OR of their knowledge vectors.

$$CD_{i,j} = \frac{\sum_{k=1}^{|K|}(AK_{i,k} * \sim AK_{j,k}) + (\sim AK_{i,k} * AK_{j,k})}{|K|}$$

$$CD_{i,i} = 0$$

Note that the CD output matrix is symmetric.

Relative Cognitive Distinctiveness normalizes each element of CD as follows:

$$RCD_{i,j} = \frac{CD_{i,j}}{\sum_{j=1}^{|K|} CD_{i,j}}$$

Thus, the elements of the ith row are normalized by the ith row sum.

Relative Cognitive Distinctiveness

Measures how distinct are two agents based on the number of knowledge bits they hold oppositely.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT:

Cognitive Expertise / Cognitive Expertise, Relative

Measures the complementarity of two agents based on their knowledge.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT: $\Re \in [0,1]$

For each pair of agents (i,j) compute the number of knowledge bits that j knows that i does not know. Then normalize this sum by the total number of knowledge bits that agent i does not know.

$$CE_{i,j} = \frac{\sum_{k=1}^{|K|} (\sim AK_{i,k} * AK_{j,k})}{(|K| - \sum_{k=1}^{|K|} AK_{i,k})}$$

$$CE_{i,i} = 0$$

Note that the CD output matrix is NOT-symmetric.

Relative Cognitive Expertise normalizes each element of CE as follows:

$$RCE_{i,j} = \frac{CE_{i,j}}{\sum_{j=1}^{|K|} CE_{i,j}}$$

Thus, the elements of the i*th* row are normalized by the i*th* row sum.

Congruence

Congruence as the term applies to the dynamic social analysts is tied to overall organizational efficiency. This measure is derived from a series of measures which, like several of the bipartite measures we learned in the previous chapters, are constantly being developed and refined at the Carnegie Mellon's CASOS Center in Pittsburgh, Pennsylvania.

We say that Congruence is mostly a "graph level" measure as it tells us something about the properties of an entity class that make up our Meta-Network. To picture this in another way, we would say knowing the connections of one particular node in a network be it to any other entity classes, would constitute an "agent level graph" whereas measures in the entire Caesar network as a whole would be a "graph level measure." In simpler parlance: agent level measures tell us about agents. Graph level measures tell us something about the entire network.

In our model of the Caesar Network we might then apply congruence measures, which make use of the three entity classes, to derive some values for the network as a whole. This could tell us how efficient the network is in terms of what skills people have in the networks and if those skills are being used for the jobs that need them the most. That is – is there congruence amongst tasks and resources and the agents that must utilize and complete them as directed?

What about our political analyst and his model of the imperial senate? What use could he or she get out of using similar techniques? Armed with three measure data, which can reveal congruence, perhaps our analyst can now get a general idea about how strong or weak one network might be compared to the next one. Would this information prove valuable in figuring out how to disrupt the network? Invariably yes.

What about our administrative advisor in charge of managing Cesar's cash coffers and resources? If he had the congruence measures applied to an optimized network, which could serve as a model for all others, could not the analyst then apply the congruence measures, or any of their variants, to his resource network? This would be likely needed to learn how strong or weak the network is compared to the standard? Would this information prove useful in understanding the bigger picture of a computer networks strengths and vulnerabilities? Once again, the answer is yes.

The math beyond congruence

Congruence, Agent Knowledge Needs / Congruence, Agent Resource Needs

The number of skills that an agent lacks to complete its assigned tasks expressed as a percentage of the total skills required for the assigned tasks.

Lee, 2004

TYPE: Agent Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: $\Re \in [0,1]$

Agent Knowledge Needs compares the knowledge needs of the agent to do its assigned tasks, with the actual knowledge of the agent.

Let N = AT*KT' = knowledge needed by agents to do assigned tasks.

for Resource replace KT with RT

We need to sum the knowledge needed but not available.

Then, Agent Knowledge Needs for

$$\text{agent i} = \frac{\sum_{j=1}^{|\mathbb{K}|} N_{i,j} * (\sim AK_{i,j})}{\sum_{j=1}^{|\mathbb{R}|} N_{i,j}}$$

Agent Resource Needs is analogous, replacing AK with AR, and KT with RT.

Congruence, Agent Knowledge Waste / Congruence, Agent Resource Waste

The number of skills that an agent has that are not needed by any of its tasks expressed as a percentage of the total skills of the agent.

Lee, 2004

TYPE: Agent Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: $\Re \in [0,1]$

Agent Knowledge Waste compares the knowledge of the agent with the knowledge it actually needs to do its tasks. Any unused knowledge is considered wasted.

Let N = AT*KT' = knowledge needed by an agent to do its assigned tasks.

for Resource replace KT with RT

We need to sum the knowledge the agent has but which is not needed.

Then, Agent Knowledge Waste for

$$\text{agent i} = \frac{\sum_{j=1}^{|\mathbb{K}|} AK_{i,j} * (N_{i,j} = 0)}{\sum_{j=1}^{|\mathbb{R}|} AK_{i,j}}$$

The equation for Agent Resource Waste replaces AK with AR, and KT with RT.

Congruence, Communication

Measures to what extent the agents communicate when and only when it is needful to complete tasks. Hence, higher congrunce occurs when agents don't communicate if the tasks don't require it, and do when the tasks require it.

Communication needs to be reciprocal.

Carley, 2002

TYPE: Graph Level

INPUT: AA: binary, AT: binary, [AR/RT]: binary, [AK/KT]: binary, TT: binary

OUTPUT: $\Re \in [0,1]$

One of the following pairs of matrices must exist: AK/KT, AR/RT. If both exist, then the measure first concatenates them into [AK AR], [KT RT] and uses them.

Communication Congruence = 1 iif agents communicate when and only when it is needful to complete their tasks. There are three task related reasons when agents i and j need to communicate:

(a) Handoff: if i is assigned to a task s and j is assigned to a task t and s directly precedes task t.

(b) Co-Assignment: if i is assigned to a task s and j is also assigned to s.

(c) Negotiation: if i is assigned to a task s and j is not, and there is a resource r to which agents assigned to s have no access but j does.

The three cases are computed as follows:

(a) let H = AT*TT*AT'

(b) let C = AT*AT'

(c) let N = AT*Z*AR', where Z(t,r) = [AT'*AR - RT'](t,r)<0

Note that C is always symmetric, but not necessarily H and N.

let Q(i,j) = [(H+H') + C + (N+N')](i,j) > 0.

Communication Congruence requires reciprocal communication, explaining the transposes of H and N to make them symmetric.

let d = hamming distance between Q and AA, which measures the degree to which communication differs from that which is needed to do tasks.

The maximum value for d is d_max = |A|*(|A|-1)

Then Communication Congruence = 1 - (d /d_max), which is in [0,1].

Congruence, Organization Agent Knowledge Needs / Organization Agent Resource Needs

Across all agents, the skills that agents lack to do their assigned tasks expressed as a percentage of the total skills needed by all agents.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: $\Re \in [0,1]$

As in Agent Needs, let N = AT*KT'

for Resource replace KT with RT

Then

$$\text{Organization Agent Needs} = \frac{\sum_{i=1}^{|A|} \sum_{j=1}^{|K|} N_{i,j} * (\sim AK_{i,j})}{\text{sum}(N)}$$

Organization Agent Resource Needs is analogous, replacing AK with AR, and KT with RT.

Congruence, Organization Agent Knowledge Waste / Organization Agent Resource Waste

Across all agents, the skills that agents have that are not required to do their assigned tasks.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: $\Re \in [0,1]$

As in Agent Waste, let N = AT*KT'

for Resource replace KT with RT

Then

$$\text{Organization Agent Needs} = \frac{\sum_{i=1}^{|A|} \sum_{j=1}^{|K|} AK_{i,j} * (N_{i,j} = 0)}{\text{sum}(AK)}$$

The equation for Organization Agent Resource Needs replaces AK with AR, and KT with RT.

Congruence, Organization Task Knowledge Needs / Organization Task Resource Needs

Across all tasks, the skills that tasks lack expressed as a percentage of the total skills needed by all tasks.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: $\Re \in [0,1]$

As in Task Needs, let S = AT'*AK

for Respource replace AK with AR

Then

$$\text{Organization Task Needs} = \frac{\sum_{i=1}^{|T|}\sum_{j=1}^{|K|} KT^{t}_{i,j} * (S_{i,j} = 0)}{\text{sum}(KT)}$$

The equation for Organization Task Resource Needs replaces AK with AR, and KT with RT.

Congruence
Organization Task Knowledge Waste / Organization Task Resource Waste

Across all tasks, the skills supplied to tasks via agents that are not required by them, expressed as a percentage of the total skills needed by all tasks.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Graph Level:

$\Re \in [0,1]$

As in Task Waste, let S = AT'*AK

for Resource replace AK with AR

Then

$$\text{Organization Task Waste} = \frac{\sum_{i=1}^{|T|}\sum_{j=1}^{|K|} S_{i,j} * (\sim KT_{i,j})}{\text{sum}(S)}$$

The equation for Organization Task Resource Waste replaces AK with AR, and KT with RT.

Congruence
Organization Task Knowledge Waste /
Organization Task Resource Waste

Across all tasks, the skills supplied to tasks via agents that are not required by them, expressed as a percentage of the total skills needed by all tasks.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Graph Level:

$$\Re \in [0,1]$$

As in Task Waste, let S = AT'*AK

for Resource replace AK with AR

Then

$$\text{Organization Task Waste} = \frac{\sum_{i=1}^{|T|} \sum_{j=1}^{|K|} S_{i,j} * (\sim KT_{i,j})}{\text{sum}(S)}$$

The equation for Organization Task Resource Waste replaces AK with AR, and KT with RT.

Congruence,
Strict Knowledge/Strict Resource

Measures the similarity between what knowledge is assigned to tasks via agents, and what knowledge is required to do tasks. Perfect congruence occurs when agents have knowledge when and only when (strictly) it is needful to complete tasks.

Carley, 2002

TYPE: Graph Level

INPUT:

Knowledge: AK:binary; AT:binary; KT:binary

Resource: AR:binary; AT:binary; RT:binary

OUTPUT: $\Re \in [0,1]$

Knowledge Congruence = 1 iff agents have knowledge when and only when it is needful to complete their tasks. Thus, we compute the knowledge assigned to tasks via agents, and compare it with the knowledge needed for tasks.

let KAT = dich(AK'*AT)

for Resource KAT becomes RAT and replace AK with AR

let

$$d = \sum_{i=1}^{|K|} \sum_{j}^{|T|} KAT(i, j) * KT(i, j)$$

for Resource replace KT with RT

let <d = d / (|K|*|T|), which normalizes d to be in [0,1]

for Resource replace |K| with |R|

Then Knowledge Congruence = 1 - d

Congruence, Task Knowledge Needs / Congruence, Task Resource Needs

The number of skills not supplied to a task, and required to do the task, expressed as a percentage of the total skills required for the task.

Carley, 2002

TYPE: Task Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT:

$\Re \in [0,1]$

Task Knowledge Needs compares the knowledge requirements of each task with the knowledge available to the task via agents assigned to it. It is similar to Knowledge Congruence, but quantifies only the under supply of knowledge to tasks.

Let S = AT'*AK = knowledge supplied to tasks via assigned agents

We need to sum the knowledge required but not supplied.

Thus,

$$\text{Task Knowledge Needs for task i} = \frac{\sum_{j=1}^{|K|} KT^{t}_{i,j} * (S_{i,j} = 0)}{\sum_{j=1}^{|K|} KT^{t}_{i,j}}$$

The equation for Agent Resource Needs replaces AK with AR, and KT with RT.

Congruence, Task Knowledge Waste / Congruence, Task Resource Waste

The number of skills supplied to a task via agents that are not required by it expressed as a percentage of the total skills required for the task.

Carley, 2002

TYPE: Task Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Task Level:

$$\Re \in [0,1]$$

Task Knowledge Waste compares the knowledge requirements of each task with the knowledge available to the task via agents assigned to it. It is similar to Knowledge Congruence, but quantifies only the over supply of knowledge to tasks.

Let S = AT'*AK = knowledge supplied to tasks via assigned agents

We need to sum the knowledge supplied but not required.

Thus,

$$\text{Task Knowledge Waste for task i} = \frac{\sum_{j=1}^{|K|} S_{i,j} * (\sim KT^{t}_{i,j})}{\sum_{j=1}^{|K|} S_{i,j}}$$

The equation for Agent Resource Waste replaces AK with AR, and KT with RT.

Congruence report for the Julius Caesar network.

These values apply to the overall network as a whole.

Graph Level Measure	Value
Congruence/Org Agent Knowledge Needs	0.47929
Congruence/Org Agent Knowledge Waste	0.193878
Congruence/Strict Knowledge	0.644928
Diversity/Knowledge	0.821949
Load/Knowledge	1.375
Negotiation/Knowledge	0.26087
Redundancy/Assignment	0.0869565
Redundancy/Knowledge	0.326241
Under Supply/Knowledge	0.26087

Chapter Summary

Now we can see that adding an even third entity class to the mix, produces even more useful possibilities from the standpoint of the dynamic network analysts in helping manage Cesar's empire. In the case of administrative advisor, we just might be able to figure out who in his organization is the real up and comer based on responsibilities and ties to others. Cesar would surely want to identify such an agent.

In the case of our military advisor, he or she may now be able to decide how strong or weak a terror cell is relative to another terror cell. They should be able to measure the skills and resources with the network to determine how it is being utilized and if it is efficient relative to a certain standard. In turn, they can now come up with a better strategy on whom to isolate in the conspiring senatorial chambers and gain a good understanding of who could cause the most disruption to the network.

Conversely, the computer network analyst has a slew of measures that can tell him more complex information because of the three entity classes now employed in analyzing his server network model.

Additionally, we learned what network congruence is and how to arrive at it using our measures which take into account three entity classes. Moreover, we learned about cognitive demand and how it is important to understanding a network as a whole.

However, there is more to understanding a network than applying three entity class measures. A powerful network analytical tool should also account for movement of the many parts that make up a Meta-Network. After all, *Whos* don't exist in a vacuum. They are in constant flux. The people or entities that comprise nearly any network are usually moving about from one place to another, which is why traditional network analysis is limiting. No sooner than a complex hierarchy of people is charted and relationships are plotted on paper, the network model is rendered useless because the people that comprise the network already moved on. They could have left the network. They could have died. This is another way of saying we have to account for how networks change over time and any model that can likely predict how they might change given certain variables would be a more valuable model indeed. Moreover, a Meta-Network that could be said to be relatively consistent in who comprises the network still must account for any one entity in the network might be at any given point in time. In the next chapter we introduce the concept of trails.

CHAPTER 5: Groups

Groups are a natural part of our human experience. Individuals are indeed social actors who tend to migrate into membership into a group, or several groups, for a variety of reasons. In prehistory ages, the human species seem to have figured out that the family group is the starting social structure for our very core survival and that being part of an even larger group provides us with increased access to critical resources, such as food, water, potential mates, etc. Moreover, being part of a group afforded the individual better protection from our enemies, both the difficulties of nature and the brutalities of other humans.

Today, this drive for affiliation continues; we each are a member of numerous groups, from personal-contact groups, to cyber-space virtual world groups. We, of course, each remain part of a family, a notion that has experienced much change in recent generations, to include same-sex marriages and divorce (or other) driven multiple and extended family-relation groups. We are each a national citizen of at least one country, which arguably provides protection. We have cub scouts, rotary clubs, alcoholics anonymous, and many more social-affiliation groups. More recently, we have virtual community groups in the form of space-warp and cyber-world online communities.

From the perspective of network analysis, groups also encapsulate objects and ideas, such as computer networks, music genres, and job categories. Whether by the force of a natural law, or the limited capacity of humans, objects and ideas are placed into groups. Groups of all types, forms and made up of various complements and, even, subgroups are everywhere.

So it follows, that a network analysis will typically investigate a network data from the perspective of the network as a group or a set of sub-groups. This perspective accomplishes two things: (1) it reduces the number of entities to analyze, and (2) the natural tendency for things (including people) to gravitate into groups, suggests that understanding the sub-groups in a network may provide hidden clues to that network.

Groups: Caesar's Organizational Restructuring

Julius Caesar is going through some major organizational restructuring. The soldiers are naturally worried but the whole idea is for the empire to realign strategically to allow the groups within the empire the most efficient use of resources. It is the theory of Julius Caesar that the empire can then become more profitable and hence more stable. Therefore, ultimately, it is about reassuring the soldiers of the empire that Julius Caesar will continue to remain strong long into the near future. Now, that should make some rest easier but perhaps it makes politicians, especially Cassius, a little uneasy. After all, they have given themselves a daunting task: they want to know about the groups that exist inside their organizational structure. They believe, quite correctly, that knowing the make up of how groups and teams function across the empire will therein provide clues as to where Julius Caesar's Empire can reallocate resources to make Julius Caesar's Empire more efficient and hence more enduring. The problem is how do they go about discovering the real groups that exist within their empire?

Should they simply ask everyone in the empire what groups they belong too? Should they ask them each individually what teams they believe they are most contacted by within Julius Caesar's Empire? Would they even know if they were part of a group to begin with? After all, what exactly is a group? How would we define a group as it pertains to Julius Caesar's Empire? Should politicians just sort of guess as to what they think constitutes a group and ask each employee if they believe this is the group they belong too? That is an approach. Is it a good one? Probably not. Julius Caesar's Empire clearly does not have an easy task (Breiger 1974).

Now another organization in Caesar's empire, this one the Military advisors is interested in understanding the dynamics of groups outside their organization. They give our military advisors analyst

the task of modeling the primary groups that exist within a vast network of enemy operatives. What should the military analyst do? Should they get the suspected enemies personal information and ascertain whether they fit into a certain group or not? Maybe the military analyst might arbitrary decide as to what a group would look like, such as those that all have a certain affiliation with a certain military commander or location along the Roman frontier? How else might the dynamic network analyst go about looking for the enemies and discovering what group they belong too? In this sense, the analyst job seems far more daunting then that of the Julius Caesar's Empire team at Julius Caesar's Empire. After all, who will cooperate with him when he asks?

What is a group?

A group specifically is a collection of things. It can be entities, nodes, ties, networks. A group might at times be represented as a meta-node, which is a collapsed collection of nodes into one entity called a Meta-Node. But, it doesn't end there. Nodes may be classified in to groups because of a shared attribute, type, id-range, label, user selection, etc. For example, if you have a set of people and know their gender, then their might be two groups - men and women. In addition, the nodes representing those people could be displayed as a meta-node for men and a meta-node for women. Nodes may be classified in to groups based on a grouping algorithm. For example, if you have a network showing connections among members of an organization and you run a grouping algorithm it will return clusters of nodes that fit together on some mathematical criteria. This cluster is a group and can be represented as a meta-node.

Fuzzy versus discrete groups

Inside Julius Caesar's Empire, we have "Cassius," who interacts with "Brutus" and "Calpurnia." Everyone interacts within his or her group. We will say that they are part of distinct organizational group called "Roman Countrymen" but could there be other groups they interact with outside of their core group? Of course there could. There could be countless amount of other groups which they interact with on a regular basis.

Cassius could hang out with the accounting group while on lunch. He may have privy to knowledge that is generally available only to those in the friend's group. Likewise, Brutus might take roman baths on the weekends and belong to the same club that many of the senior leaders belong too. He may therefore have the inside scoop on a lot of high-level strategy that generally would be unavailable to anyone within the customer service group. He may even serve as a source of information to the senior executives as to what really works in the costumer service group.

In terrorist circles, the Military advisors may be interesting in knowing how such terrorist members leave their individual cells and communicate with other around the globe. He or she may be interested in exactly what other groups they are a member of by association. The thinking is that people, agents, belong to more than one group, and often do. Those members constitute fuzzy groups – groups composed of people that otherwise do not have direct affiliation with it. These can be critical to reveal in terms of a terrorist network (Davis and Carley).

For instance, it could be that a particular enemy cell may appear to be benign. The analyst may be lulled into a false sense of security. What if the other groups that any particular enemy within a group communicates with appear to be benign as well? And that may well be the case. However, if you consider the members of the cell that interact with other cells outside of their cellular environment, the analyst might get a wholly different view on how active a group might appear to be. The fuzzy group could be highly active and highly dangerous ready to strike at any moment, sharing ideas, networking resources and building knowledge. These groups would remain hidden under traditional network

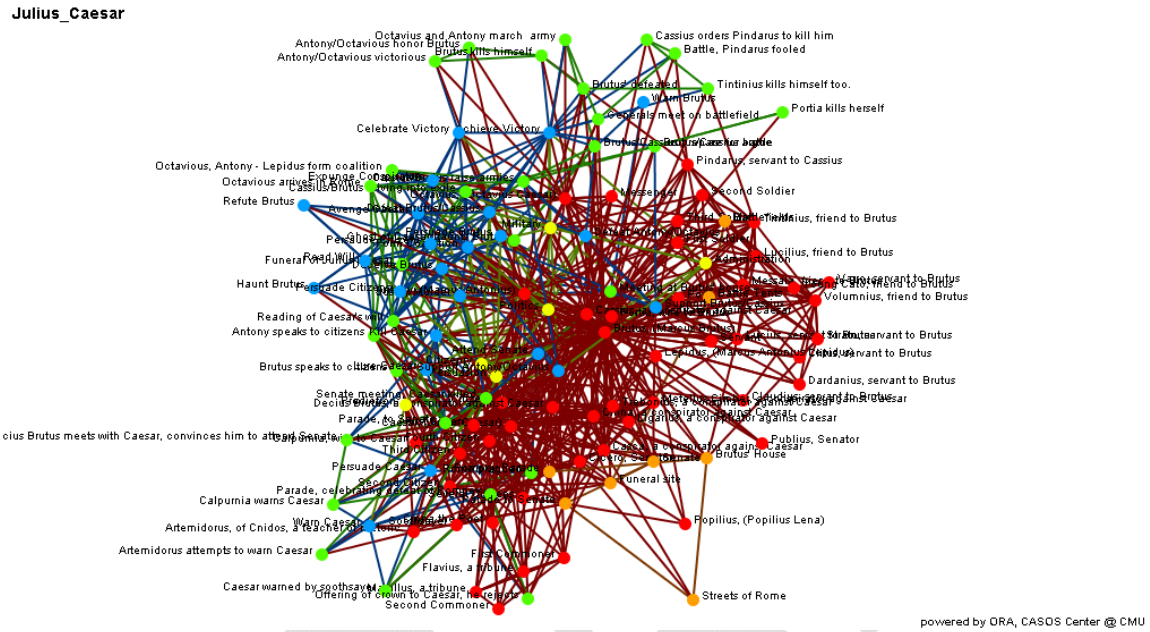
analytical methods. The Dynamic Network Analyst would find them and appropriately assess them. Julius Caesar would be happy.

Finding discrete groups

The Group Viewer helps separate entities into distinct groups. Take the starting view from the visualizer. Not much can be discerned from this rather jumbled view with all of the entities chosen. There are five groups which can be called up: Concor, Newman, Johnson, Fog and K-Fog. Each one of them is base don a mathematical formulae that has proven useful to the network analysts in studying networks. This is to say they group according to slightly different criteria but criteria that are highly useful.

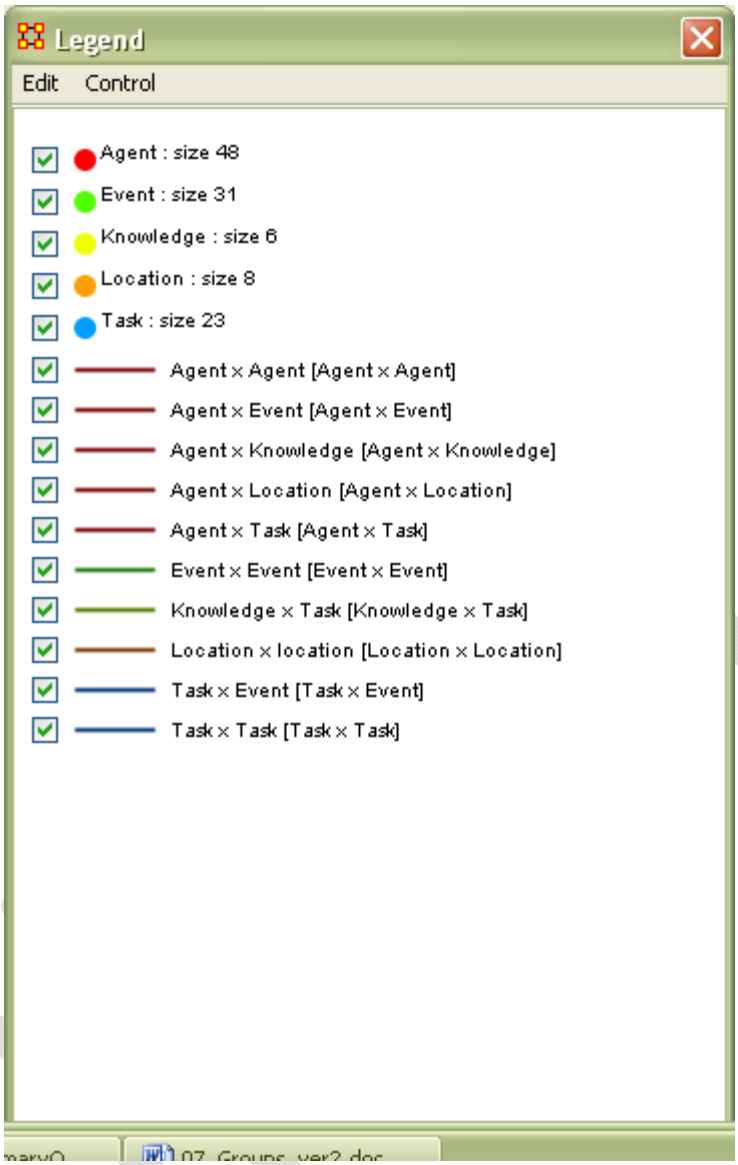
Finding groups with ORA

One software tool used to find fuzzy groups and discreet groups *ORA. Using this tool, we will extract group information for a Meta-Network based on Julius Caesar dataset. In this particular example, we are using a Meta-Network that would be of particular interest to the military advisors:



When you first call up the Visualizer, every entity is listed and displayed on screen. At first glance, not much can be discerned.

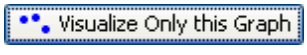
One way in which the information makes much more sense is to visualize one of the graphs, which as you may recall, can be a relationship such as Agent x Agent, Agent x Knowledge, Resources X Agent, etc.



ORA allow you to highlight one of the graphs you want to view in difference groups.

Figure 21: Drop down menu from ORA: Organizational Risk Analyzer

Then Select Visualize Only this Graph



We now have a much more manageable display.

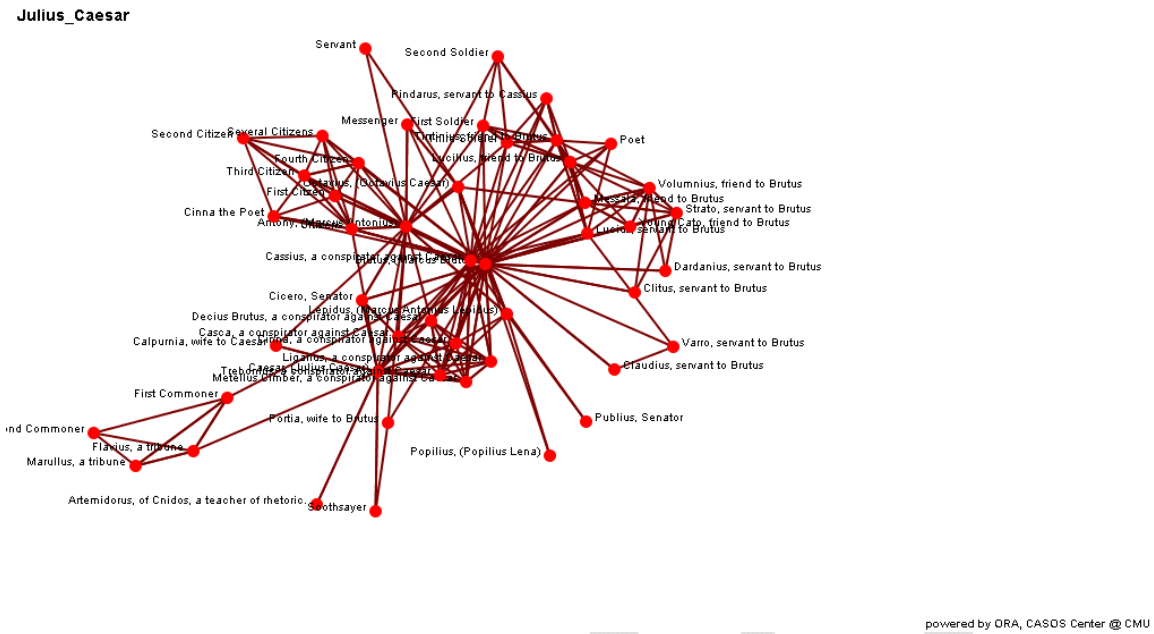


Figure 22: Agent by Agent Visualization

ORA allows an analyst to choose the type of group we wish to view. Here the group viewer tool is highlighted.

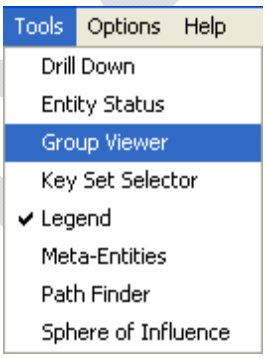


Figure 23: Tools available in ORA

Which calls up the Group Viewer dialog box. First let's see what happens with the CONCOR options. Select [Compute] to activate the grouping.

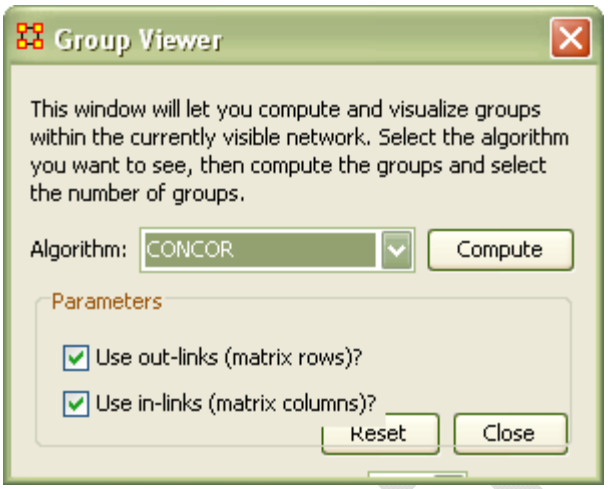


Figure 24: In this example the Concor grouping algorithm is selected

The default is one group with all visible entities connected to the center

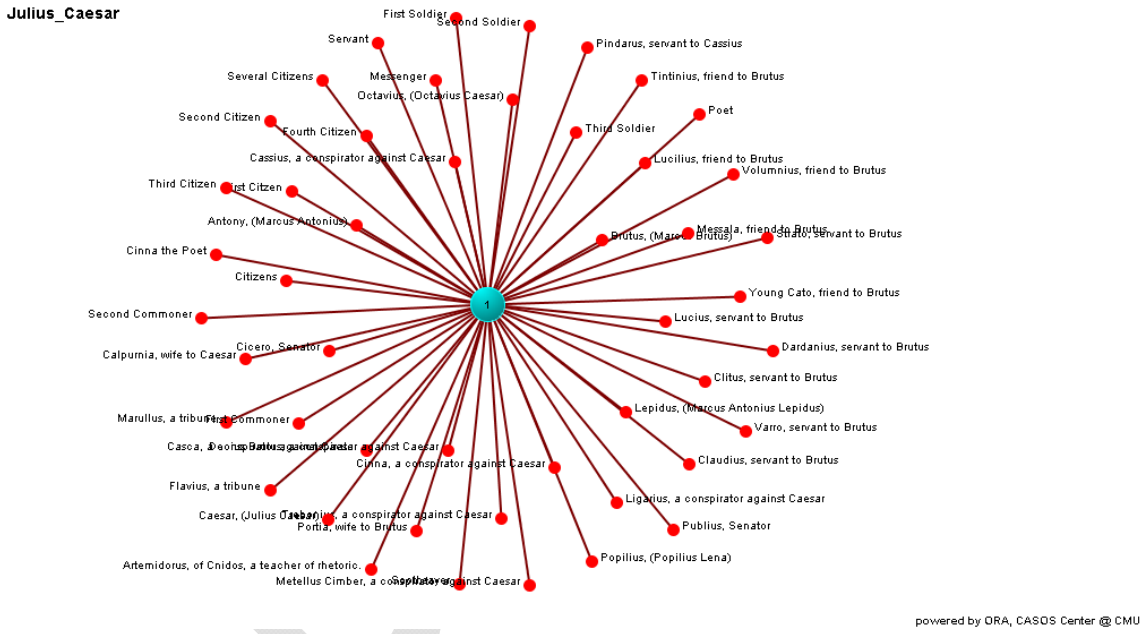
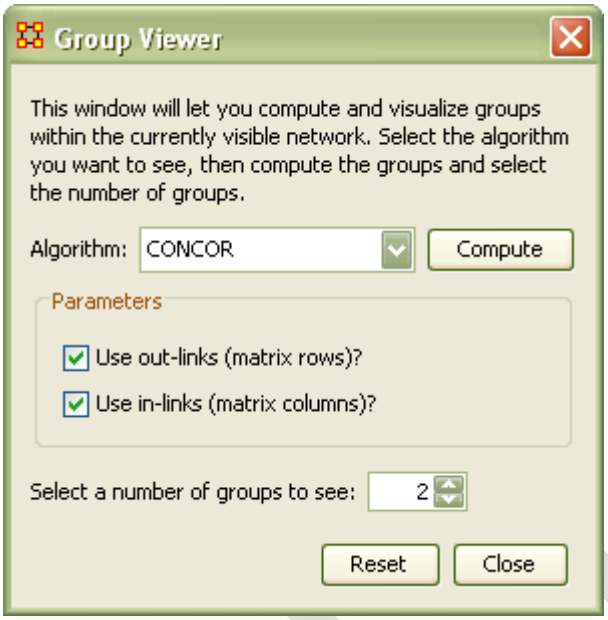


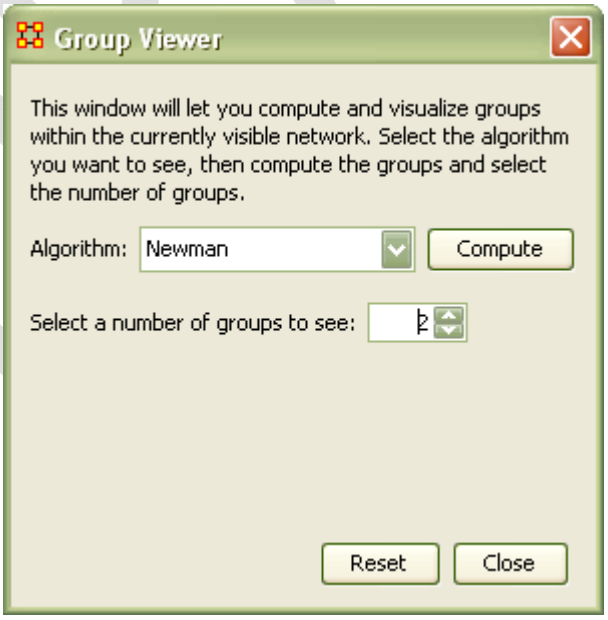
Figure 25: ORA creates group based on Concur mathematical formula

The dialog box, increase the Select a number of groups to see: to 2.

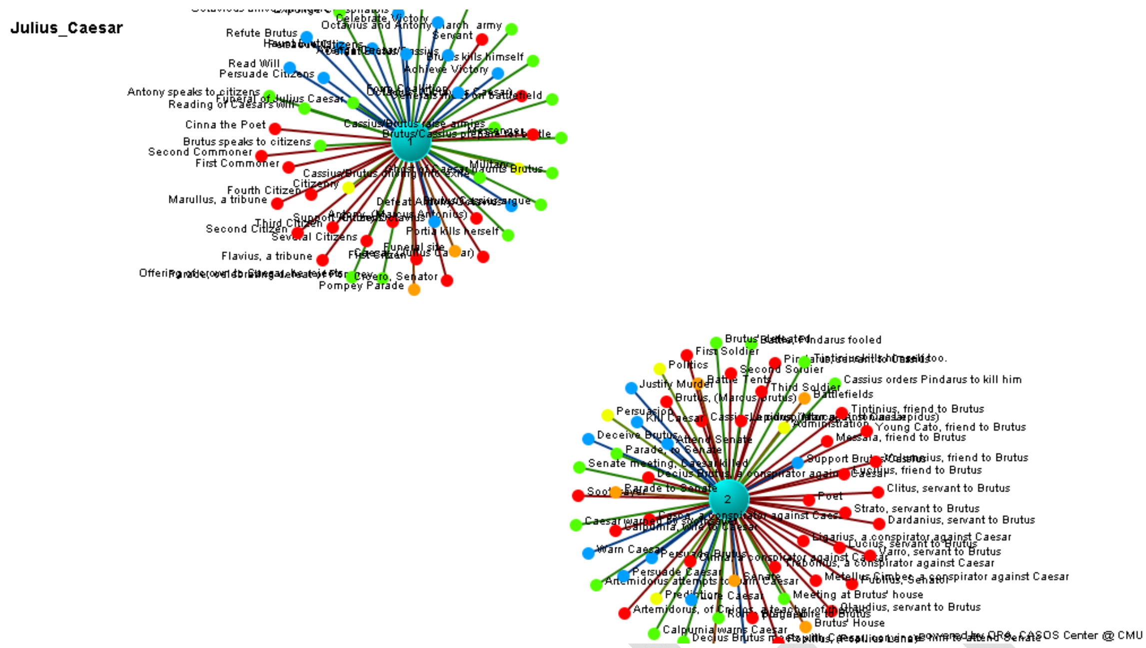


Now displayed are two groups.

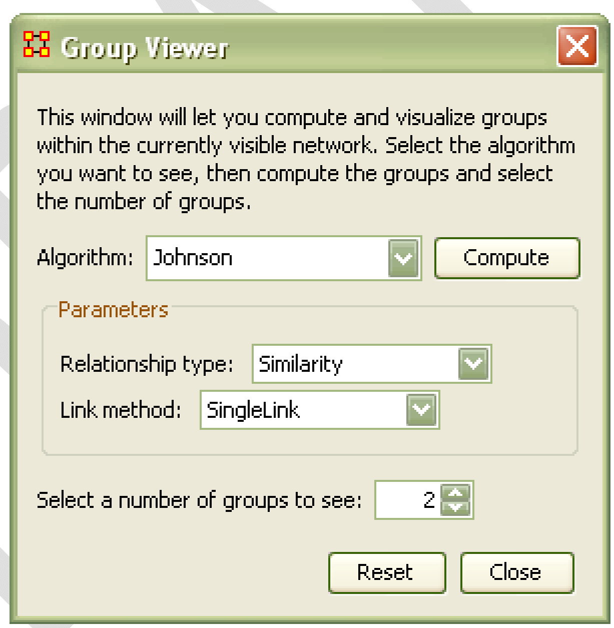
Now we will change the group to "Newman" and select and we can increase the number of groups to see to "2".



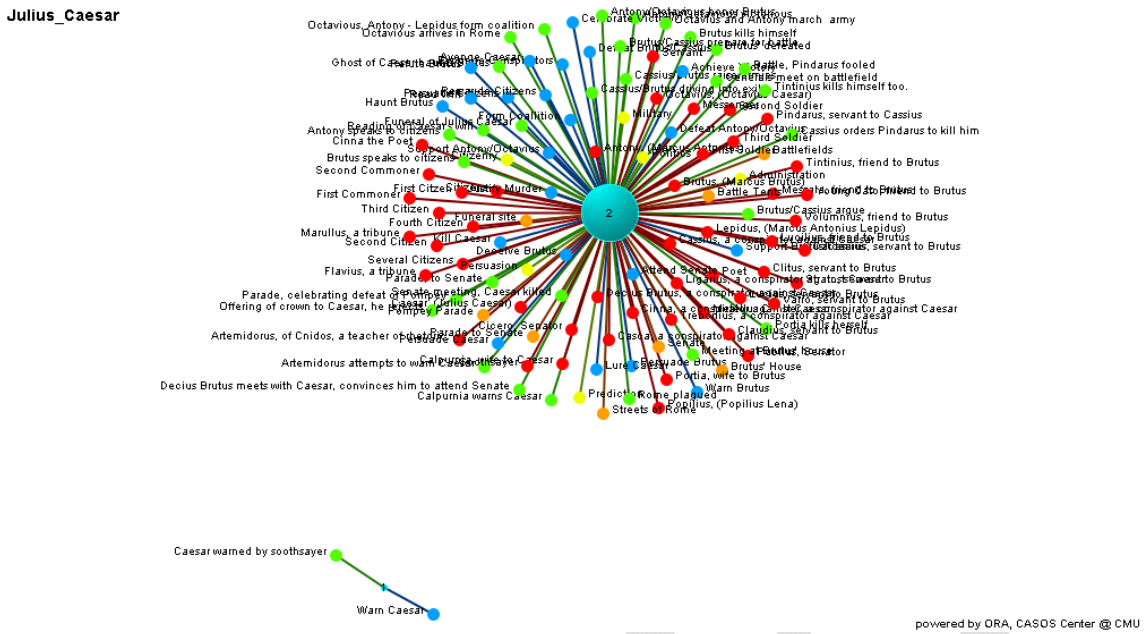
Once again, we see two groups but the division is different. The left group contains all the knowledge entities except driving expertise. And now there are fewer agent entities connected to this knowledge.



Now change the group to “Johnson”, select, and increase the number of groups to see to “2.”



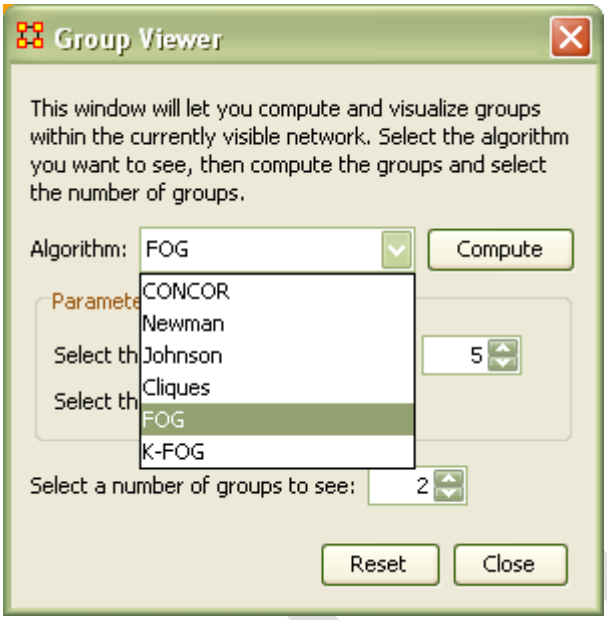
It is still the only knowledge in the right group but notice that now there is only one agent connected to it, mustafa_fadhil. Looking back, he was connected to driving expertise in Newman but not CONCOR.



Finding fuzzy groups

The ORA Visualizer can locate FOG Groups (Fuzzy Overlapping Groups) within your MetaMatrix. FOG indicates that entities can belong to more than one group in varying strengths and the likelihood that those entities will participate in events associated with the groups it is connected to in other groups. We can see the importance of this to Julius Caesar. After all, it would be helpful to know not only what groups someone like say Cassius is in that they both have in common, but what other groups does he belong to as well. To access the ORA Compute Fog Groups you must first be working in the Visualizer. You will be presented with the normal visualizer display. Before you start, it is a good idea to Show Edge Weights. This will assist you in seeing more clearly the strength of the connections from each entity to each group (Davis and Carley).

From the Visualizer menu select Option > Show Edge Weights
Then from the Visualizer menu select Tools > Group Viewer



After the Group Viewer appears select FOG form the drop-down menu. Then select the [Compute] button.

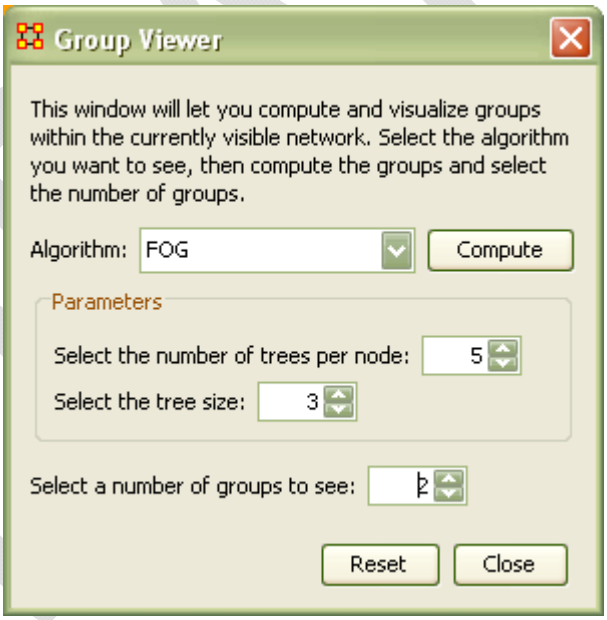


Figure 26: Choosing Fuzzy Overlapping Group in ORA

The display starts out with all agents connected to one single group. You'll notice that some edges are weighted heavier than others.

Figure 27: Weight thickness shown in ORA indicates strong ties

Under Select a number of groups to see: increase the number to 2 groups. For this example let's concentrate on ali_mohamed. At 2 groups he is connected to both of them. Thusly he still has contact with the entire network. His connection with the group on the left is stronger than his connection to the group on the right.

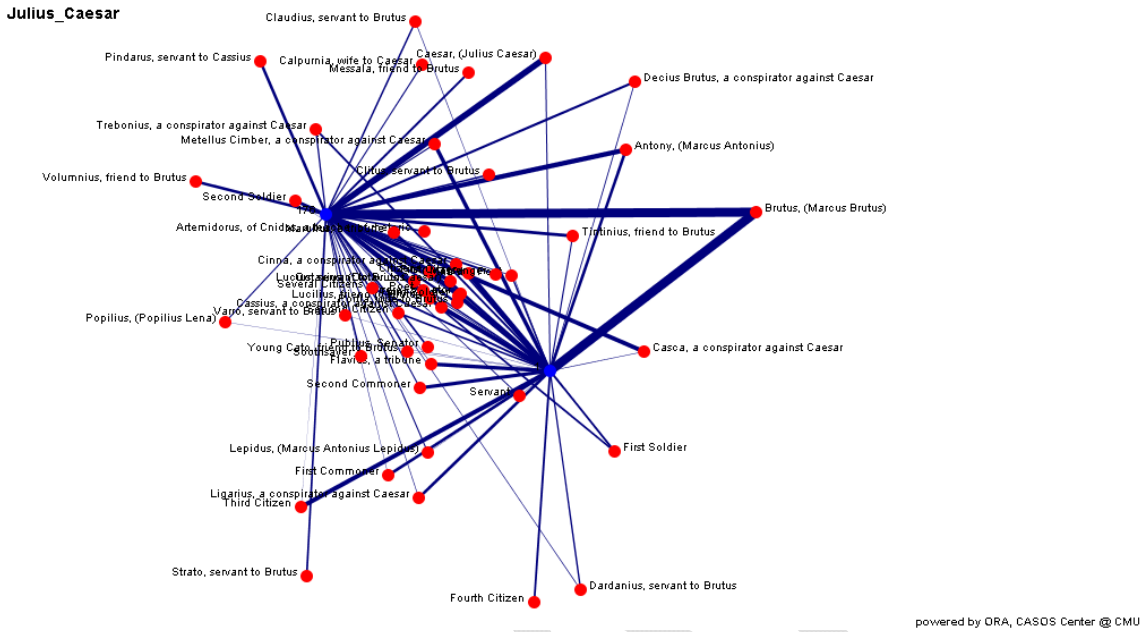
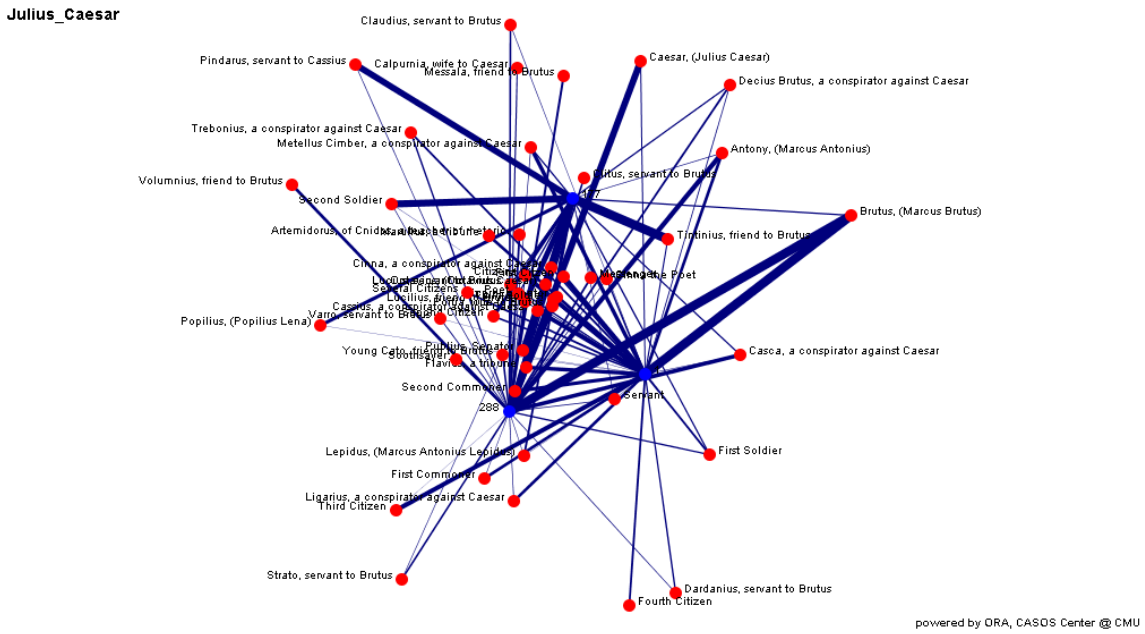


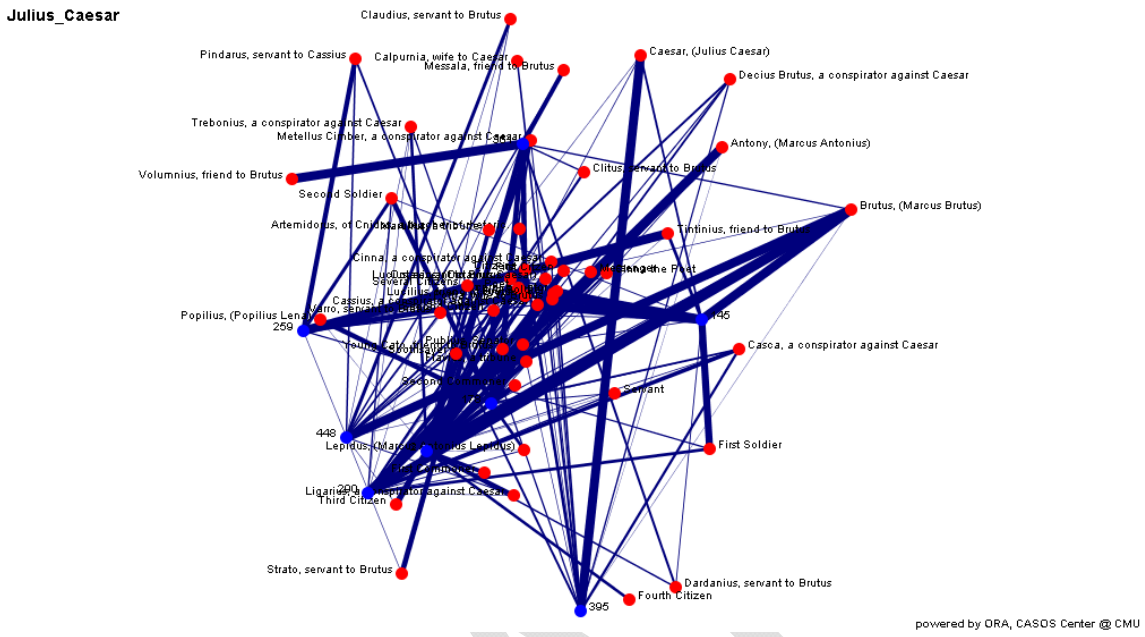
Figure 28: Agents with overlapping ties shown in ORA

Now let's increase the display to "3" and observe the results.

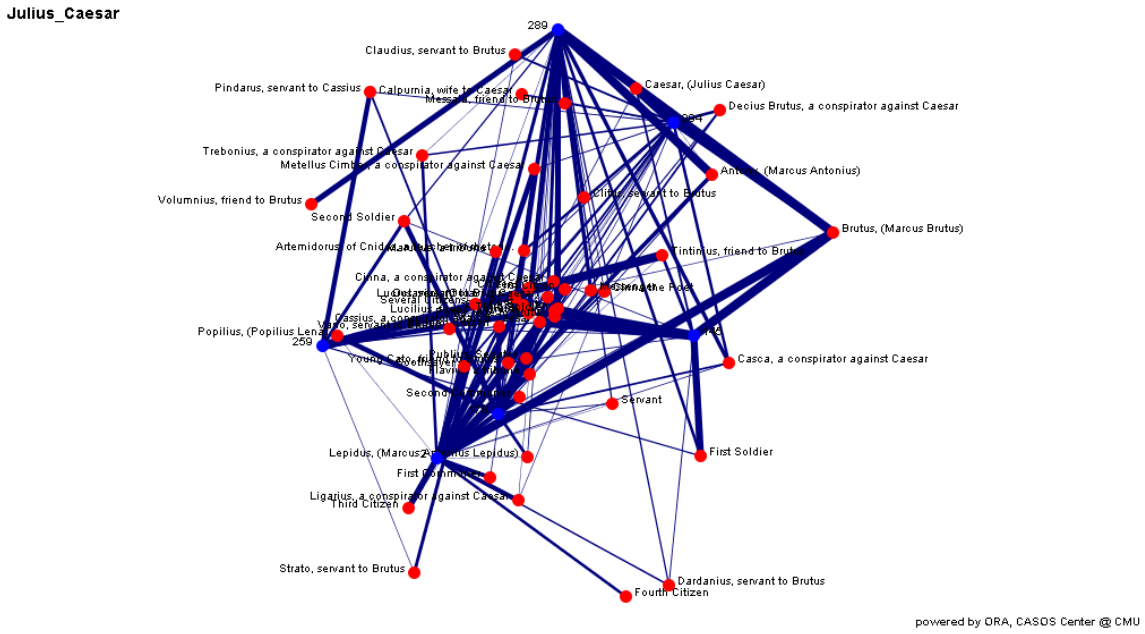


After an increase to four groups, you can see that many nodes are still connected to the same people as before but the grouping is again changed and relevance is added to the other new groups created by the

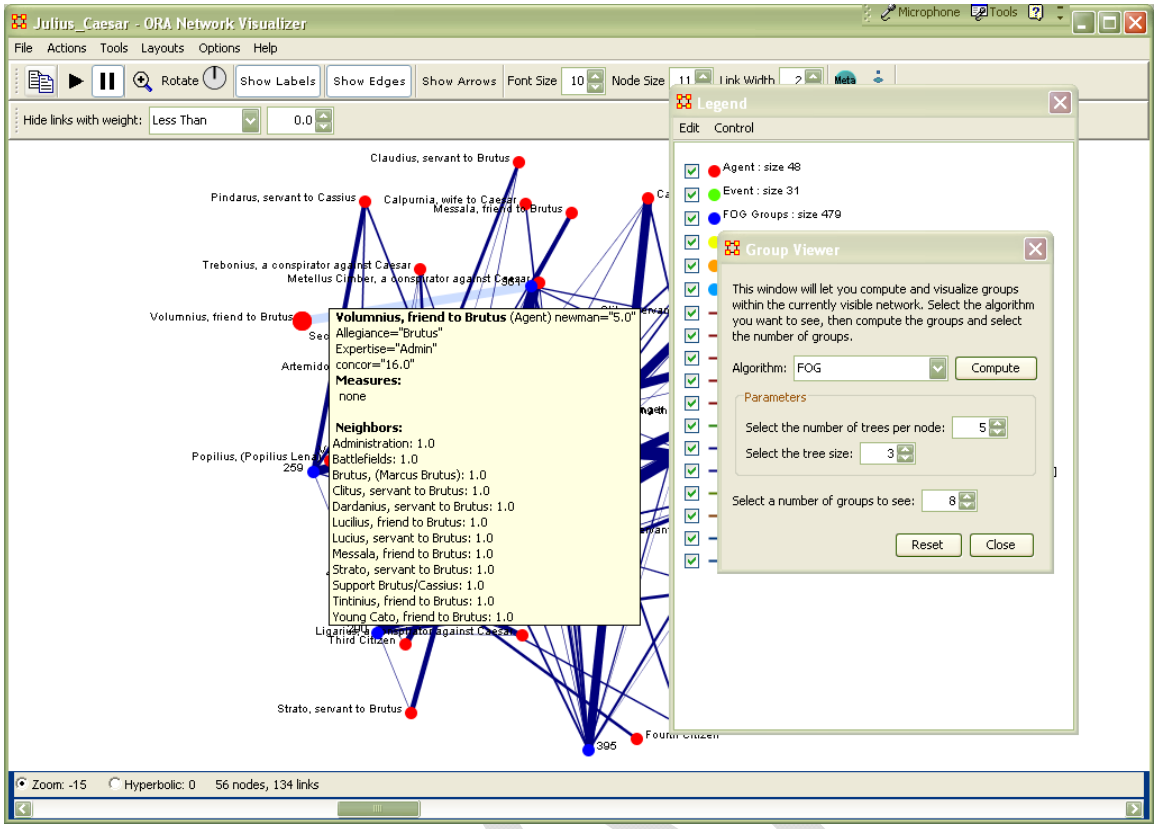
algorithm. You'll notice that the strength of the tie to the group in the lower left is the strongest connection and he has no connection to the group in the lower right.



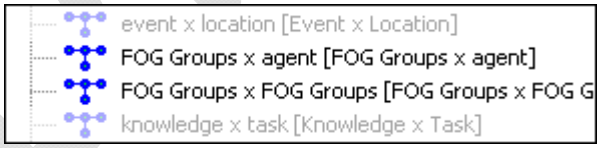
Our final increase to 8 groups shows little change to the groups. It is then we must start drawing conclusions about how many Fuzzy groups we are interested in seeing.



By left-clicking on any entity or group you bring up the Entity Status dialog box from where you can retrieve information about its attributes, measures, and neighbors.



After you run the the FOG Group viewer there are Entity Sets and Graphs that are created in the MetaNetwork. These will be displayed in any future work in the Visualizer. If you do not need those after you are done with the Group Viewer then don't forget to delete them. Deleting the Entity Set for the FOG group will delete any associated Graph.



Finding groups on multi-mode data

Location of critical individuals, groups, technologies given any network, such as a communication network, or alliance structure, or monetary flow, where the nodes are individuals, groups, computers, etc., a number of network measures such as centrality or cut-points can be used to locate critical nodes. Additional measures based on an information processing view of organizations also exist for locating critical soldiers, redundancy, and potential weak points within groups and organizations.

Many of the traditional social network measures and the information-processing network measures are embedded within ThreatFinder is a computer program that uses a combination of network analysis and multi-agent modeling to determining the potential information security risk from personnel that an organization faces due to its architecture. The degree, type, and location of possible threats, such as critical soldiers and lack of redundancy are assessed. These "location" techniques are useful within companies to help ensure information security and are useful within and among groups and organizations in mitigating the effectiveness of networks.

For example, individuals or groups with the following characteristics can be identified:

An individual or group where removal would alter the network significantly; e.g., by making it less able to adapt, by reducing performance, or by inhibiting the flow of information.

Illustrative nodes are those high exceptionally high in centrality or high in structural holes.

1. An individual or group that is unlikely to act even if given alternative information. This can be found as an individual high in centrality and Simmelian ties.
2. An individual or group that if given new information can propagate it rapidly. Such individuals may be seen as gossips, innovators, or early adopters. Possible indicators are high degree centrality or high structural holes.
3. An individual or group that has relatively more power and can be a possible source of trouble, potential dissidents, or potential innovators. Individuals with relatively more power may be high in centrality.

1. Possible innovators may be those who are isolates or those who have moved about so much that they have broad and distributed knowledge and contacts.

2. An individual or group where movement to a competing group or organization would ensure that the competing unit would learn all the core or critical information in the original group or organization.

3. An individual, group, or resource that provides redundancy in the network. Measures of redundancy are available in ThreatFinder (Carley, 2000). For the measures discussed above most can be calculated using UCINET3 or the meta-network

A last word on Caesar's Restructuring

Julius Caesar's Empire, having compiled Meta-Network Data, hands over their dataset to the Dynamic Network Analyst who then runs the gamut of grouping algorithms on it. What is revealed is an even more complex structure that is beyond the perception of the computer unassisted mind. We have the bigger groups, we have hidden groups, we have groups that some members of Julius Caesar surely didn't know existed. How do those groups function? Who are the most important people in them? Why are the groups important? How did the groups form? Where are the groups located?

The DNA analyst can answer those questions. He or she can run the measures we have learned about on them and provide statistical profiles of such groups, what their weaknesses and vulnerabilities are as well as their strengths.

It should be apparent the value such group data has to the military analysts. He or she is revealed an intricate picture of how the cellular network operates and interacts at the group level. This information

can be crucial to the dynamic network analyst in providing detailed information to key decision makers whose job it is to disrupt enemy networks.

Therefore, we hearken back to our military advisor. From the schematic level who uses the server network and in what groups they belong could be of critical importance to allocating resources to make the server network operation as efficiently as possible.

As we move into the next chapters, we will discuss how Groups fit into the context of detecting network change, how we measure change and when change is significant to measure in terms of evolving networks. We will also discuss how the Dynamic Network analyst can run simulations on networks in the near term and long term aspect.

GROUPS

As the Caesar story show, sub-groups are an important and powerful perspective when studying an organization. However, when working with sub-groups in the network setting, there is not yet agreement on a precise definition (Seidman & foster 1978; Alba & moore 1978; Mokken 1979; Burt 1980; Freeman 1984; Freeman 1992; Sailer & gaulin 1984). Exactly what constitute a group is a matter of one's perspective, the data, and analytic question; deciphering who is a member of what group remains an imprecise art, though there are some rules of thumb that experienced analysts use, these guidelines are fluid and are called into the process according to the analyst's experience and skill set. For our purposes here: cohesive sub-groups, or just "groups", are collections of nodes that share some specific characteristic(s) or property(ies); most often a node's membership is bounded by being in only one sub-group, but some grouping processes allow for multiple memberships. We seek to identify and characterize these groups because often the analyst can locate, often hidden, sets of actors that have something (loosely defined) in common that can be useful information in explanation, understanding and forecasting, either the group itself or the larger network. As mentioned earlier, these often-hidden groups are sometimes (self-) organized for come collective action of sorts. We also explore groups in order to confirm that the data collected is consistent with the observations and expectations that are already known about the group or larger network, e.g. does the social network in a department coincide with the structure of the reporting or task network. To reduce the data being analyzed; this operationalized by collapsing each individual group to a individual node; thus reducing the absolute number of nodes that form the network. This is, at times, at the cost of information lost, however. We have found that we can characterize and understand the greater network from understanding its sub-group parts. Moreover, behavioral researchers have found that a network with two subgroups differs from a network with 20 subgroups, and that the amount of overlap in group memberships have an effect (freeman_2000_2260). From the individual perspective, where an individual lies within a subgroup is indicative of their opportunities and constraints, e.g. their relative distance from a demographic fault line can affect one's self-identity (lau_2005_11234).

As we know, a network is made up of entities and relationships among those entities. Along with these webs of relationships, frequently, the entities are known to each have one or more attributes that can differ among the entities making up the network population. Using these two valuable pieces of information, we can identify subgroups according to attributes (similar or dissimilarity) of each node, and/or the characteristics of the relationships that each node has in the particular network.

There are five grouping techniques in ORA that are heavily depended on by network analysts: (a) CONCOR, (b) Newman, (c) Johnson, and (d) Fog / K-Fog. Each, by way of its approach to the problem of locating groups has its strengths and weaknesses. Which technique to use is a matter of the combination of the research question, the make-up of the data itself, and analyst experience. CONCOR puts its focus on the notion of structural equivalence to discriminate among the nodes to form groups. Structural equivalence is the notion that two nodes that have the same number of ties with the same alters are therefore structurally equivalent. CONCOR essentially perform often multiple, row or column-wise,

vector correlations to determine the level of structural equivalence between a given pair of nodes. This correlation process is repeated until ultimately the matrix representing the network has stabilized with a set of 0's and 1's. These values indicate to which group an individual node will be placed. Notice, this situates CONCOR to creating only up to two groups. CONCOR can repeats itself multiple times to further split one or both of the previously located groups. Therefore, CONCOR most often produces a number of groups that are a power of two. The number of groups identified by CONCOR is a user-parameter (often times the number of "splits" is the expected input).

The Newman technique takes a different approach. Instead of focusing on structural equivalence, Newman look at the existing ties in the network to locate a tie or the fewest number of ties that when removed will best separate the network into two (or more) components, and thus creating sub-groups. This often times, is a more natural way that we look look at our groups in an organization. We tend to think of two different groups as those that are not tightly connected and by searching for "bridging" ties and removing them, Newman is effectively performing the same task that we tend to do naturally.

The Johnson procedure uses a distance metric to discriminate groups. It creates groups according to a network that is constructed with ties that indicate the distance between dyads. Johnson will segregate groups according to these weights by separating those who are most distant from others in the same group from the original group and off to a group of nodes that are closer in this distance value.

The FOG / K-FOG are powerful approaches that recognize that nodes can often be members of more than one group at a time (fuzzy groups), which can be a major weakness of the aforementioned techniques. FOG begins with a collection of nodes and uses a maximum likelihood perspective to determine the probability that a tie exists among the various dyads in the group. It compares the probability with the actual data and as a result, a node can indeed be assigned membership in the group, or not. By taking this approach, FOG can assign a node to multiple groups.

The best grouping technique to use is not a mechanistic choice. Often times, analysts try the various approaches and simply depend on the one that produces the groups according to what makes sense. This area of network analysis is still very much a work in progress as new and complex mathematical and machine learning techniques are increasingly being applied to this aspect of network analysis.

CHAPTER 6: Analyzing Spatially Embedded Networks

1.1.Goal

In exploring the various influences and roles in human relationships we will now consider one of the most important features for understanding large-scale social interactions: space. Clearly, space is important, and as such, we intend to equip the reader with appropriate tools for analyzing networks where locational information is available.

This chapter should provide an overview of the core issues in analyzing spatially embedded networks, especially the importance of aggregation and the resolution of analysis. Key concepts include representation, aggregation & clustering, information loss and smoothing.

We proceed with an overview of the relevant social theory. Section 3 describes the tools and datasets you will use, continuing in section 4 with the core methodological concepts. In section 5, we provide a step-by-step tutorial, concluding in section 6 with some sample problems.

2. Social Theory and Background Information

2.1.Propinquity

Physical proximity has long been known to play a major role in shaping human interpersonal relationships. In general, people are more likely to interact with others who are nearby. This effect, called propinquity, has been documented time and time again for a wide variety of networks(Butts2002;Faust2000;Festinger1950;Latane1995). It has even been proven theoretically that for large social networks, the locations in space of individuals can explain almost all of the information in the network(Butts2002). Although this phenomena of propinquity is important in understanding social networks in general, it does not necessarily aid us in the analysis of any single network in particular.

2.2.Spatial Networks vs Spatially Embedded Networks

More recently, work has focused on the methodological issues regarding the analysis of *spatially embedded networks*(Davis2008b). Spatially embedded networks are networks of any entities and relationships where some (but not necessarily all) of the entities are associated with some positional information(. Spatially embedded networks are distinctly different from *spatial networks* which are networks describing the relationships between locations. Spatial networks have been examined in the geographical information sciences for many years, primarily with respect to storage/retrieval(Sankaranarayanan2005;Papadias2003) and with the discovery of shortest paths(Kolahdouzan2004). Spatial networks are a subset of spatially embedded networks with a number of additional constraints. First, entities in spatial networks are limited to locations. Second, locations are unique. These two constraints highlight one of the more interesting problems in examining spatially embedded networks, that multiple entities may exist at the same location, and that co-located entities may have differing relationships. Because of this, one potentially useful development in spatial networks, flow visualization (Phan2005;Post2003) is much more complicated with spatially embedded networks. The underlying problem of co-location surfaces often in visualizing and analyzing spatially embedded networks.

2.3.Continuous Space vs Discrete Networks

The incorporation of spatial locations into network data requires dealing with a fundamental disconnect between spatial information and relational information. Spatial information is fundamentally continuous, whereas, relations, in contrast, are defined as existing between pairs(sets) of discrete entities. This presents a barrier to the unification of spatial analysis and network analysis. Although we can break geographic space down into discrete locations(Phan2005), it is ultimately a continuous dimension, and

any partitioning necessarily results in lost information. This loss of information increases the risk of the ecological fallacy(Robinson1950) and the related modifiable areal unit problem(Openshaw1981).

The ecological fallacy is based on the observation that any calculation on aggregated data carries the risk that subsequent results may be an artifact of the aggregation. This happens because once data are aggregated, any subsequent analysis assumes that the data within the aggregate unit are homogeneous(Robinson1950;Openshaw1981;Openshaw1999) and any individual differences are unimportant. Consider the example dataset show in Table 1 and the aggregations shown in Tables 2 and 3. Although both aggregations obscure the information in the original data, the moderate aggregation better approximates the original data than the extreme aggregation. This is an open problem, but one approach to minimizing the risk associated the ecological fallacy is to explore a wide variety of different levels and methods of aggregation. Results that are consistent regardless of the specific aggregation are more likely to be due to the properties of the actual phenomena. We will explain how this principle can be applied to analyzing spatially embedded networks in section 4.

2	5	6	2
3	5	5	1
4	5	2	3
7	4	1	2

Table 1: Original Data

2.5	5	5.5	1.5
5.5	4.5	1.5	2.5

Table 2: Moderate Aggregation

4	4.75	3.5	2

Table 3: Extreme Aggregation

1 to 2 pages – on what is the history, core ideas, theories, concerns. Most citations to papers will go in this area. Note make sure you site 2 to 10 of the most key items in this area. Do not feel you need to cite everything. The items cited should be the canonical must reads, and items from the CASOS group if relevant. Of course, it would be nice to think the CASOS material was a "must read."

3. Tools and Data

For this chapter, we will leverage the Ora suite of network analysis tools and particularly Ora-GI (Ora with Geographical Information). Although the Ora-GI tool can be used with a wide variety of datasets, we will use the Tanzania dataset (see Appendix) for our examples and tutorial.

4. Methodology

In this section we will introduce the basic techniques for analyzing spatially embedded networks as well as some methods for resolving the theoretical issues described in section 2.

4.1. Representation

So far, geographic information has been described as abstractly associated with the entities in a network but there are two different ways of associating locations with the entities in a network. First, positional information can be represented as attributes of the entities in the network. This is how information is common represented in geographic information systems. In many networks, however, it may be preferable to represent location information as "located-at" relationships. For example, the city of Pittsburgh may be one entity in the network. It has associated with it some locational attributes, in this case, latitude and longitude. There are also several people in the network, John and Mary. Instead of giving John and Mary location attributes, we connect both John and Mary to the Pittsburgh entity, indicating that John and Mary are co-located in Pittsburgh. This schema is, in fact, quite useful, where one (or more) set of entities are locations with positional attributes and all other entities link to those locations rather than have their own positional attributes. Among other things, this greatly simplifies both dataset maintenance and co-location analysis.

4.2. Visualization

Most network analysis begins with visualization and the analysis of spatially embedded networks is no different. Visualizations ``have provided investigators with new insights about network structures and have helped them to communicate those insights to others"(Freeman2000). Although the visualization of complex multi-mode spatially embedded networks is often not as clear and instructive and simple single-mode network visualizations, there is still much that can be gained from them.

In addition to simply projecting a network onto the map, positioning nodes on a map where they are located, we can also leverage network measures in a manner similar to traditional network visualizations. In particular, we can "color by" and "size by" various networks analytical techniques. In general, as with traditional network visualization, it is usually most useful to "color by" a network grouping and to "size by" a network centrality measure.

4.3. Summarizing Network Information by Location

This type of visualization is somewhat complicated when displaying a network on a map due to the fact that multiple entities may be co-located. Because of this, it is necessary to summarize the network analytical information of the entire location when displaying the network on a map. For a grouping algorithm, it seems reasonable to simply assign a location to the group that the majority of entities at that location belong to. Although some variety of proportional assignment to groups may be more desirable, a simple majority assignment has been implemented in Ora-GI

For network centrality measures, it is slightly more complicated. Although, there are many possible ways of performing this summarization, a simple summation makes intuitive sense for a number of network measures, especially betweenness centrality and eigenvector centrality. For example, standard betweenness centrality counts the number of shortest paths that go through the ego node. The sum of the betweenness centralities for all nodes located at a given location is then the total number of shortest paths that go through that location. Alternatively, consider eigenvector centrality. One interpretation of eigenvector centrality is as the likelihood of passing stopping at the ego node on an infinite-length random walk through the network. Then if the eigenvector centrality of a node is the probability of stopping there, then the sum of the eigenvector centralities for all nodes located at a given location is then the probability of stopping at a given location on a random walk through the network. Summation does, however, have some undesirable properties. In general, it places much greater emphasis on locations with

a large number of nodes located at them. For example, given two locations, A and B, if location A has a few extremely important individuals and location B has many unimportant individuals it may be that using summation to summarize the locations, location B will appear to be more important.

4.4.Limitations of Visualization

Although these visualizations are useful, the projection of the network onto space reduces the flexibility of the visualization. In particular, one way in which visualizations yield useful information is through the use of layout mechanisms designed to highlight topological properties of a network, such as centrality, cohesive subgroups, etc. The most straightforward way of visualizing spatially embedded networks is a simple projection of the observed nodes onto a map based on their observed locations. By a priori choosing to locate nodes according to their observed spatial attributes, we lose any opportunity to arrange the network according to its topology. This makes visualization of spatially embedded networks less effective in illuminating and communicating the network structure. Furthermore, experience with real-world networks suggests that a small amount of noise or background activity can further decrease the utility of these visualizations. In large, noisy networks this simple projection may not yield an effective visualization. Although seemingly undesirable, one way of improving the visualization is actually to hide the edges in the network. Although this simplifies the visualization, it means that the only network information presented is any networks analytic "color by" or "size by" visuals.

4.5.Aggregation and Information Loss

The risk of committing the ecological fallacy can be reduced in spatially embedded networks by combining density-based clustering with feedback regarding information loss. The ecological fallacy, described in section 2, occurs during the analysis of aggregated data when faulty conclusions are drawn because of the specific aggregation chosen. One way of reducing the risk of committing the ecological fallacy is to examine the data at multiple different levels of aggregation. Ora-GI facilitates this through a dynamic density-based clustering algorithm, called DBScan(Ester1996). DBScan is very similar to a spatial join, except that it introduces the concept of *outliers*, points distant from any neighbors, as distinctly different from clusters of points that are joined together. Users can directly control the distance at which this join/clustering is performed and then see the impact on the network visually.

In addition to facilitating multiple levels aggregation, Ora-GI provides user feedback regarding the information lost in any individual aggregation. Information entropy is a mathematical formalization of the amount of information needed to describe a system of random variables. The information entropy of a system is equal to the expected number of bits required to communicate the current state of the system. In a network, the information content is the set of edges in the network. Given some specific assumptions about the distribution of edges in the network[1], it is possible to directly compute the information entropy in an single aggregation of a network. By comparing two different aggregations, you get a quantity called the *information loss* which represents the mathematical quantity of information that was lost due to aggregation. Interpreting that raw information loss number may sound daunting, and it would be, but if we instead compare any single aggregation (information entropy value) to both the maximum (gotten from aggregating the network into a single node) and to the minimum (no aggregation), we can get the fraction of the total information in the network that was preserved/lost by the current aggregation.

1 Edges from node u to node v are independent identically distributed random variables from a Beta distribution with certain probabilities on the parameters of the distribution.

4.6.Smoothing and Interpolation

In addition to visualizing these network analytics as distributed across individual locations, we can also use interpolation techniques to attempt to infer how the network structure is distributed across space. By using these techniques we can start to distinguish consistent spatial patterns from non-spatial effects. In addition, these can reduce the visual clutter, making it easier to analyze and communicate the analysis of large datasets.

Many different techniques exist for the smoothing and interpolation of discrete observations over space. However, not all them are appropriate for this application area. For example, kriging works best with an even sampling of the space, which is unlikely, and manual inspection of the variogram, which is undesirable (Cressie1993). Because both the locations[2] and network statistics may be noisy, we use a robust, general purpose kernel-based method.

One technique for visualizing large spatial data sets uses kernel density estimation to interpolate the point intensity across the spatial region of interest. Kernel smoothing uses a kernel function, k, to interpolate the intensity, ¸ , of a phenomena based on the observed set of discrete observations. For a target location, s:

¸(s)=∑i:D(s,si)· ¿ 1/(2¿2)k(D(s,si)/¿)

where ¿ is a bandwidth parameter and D(s,si) is a distance function. The bandwidth parameter, ¿, represents the fundamental tradeoff between a smooth interpolated function and a loss of information and oversmoothing. Although the choice of a kernel function may appear to be a difficult and important decision, it is considered to have relatively little impact on the interpolated function(Waller2004). This intensity estimation procedure is frequently used to create heatmap-style images but a variation called kernel smoothing(Hastie2001).

Kernel smoothing expands this method to smooth values rather than intensities(Hastie2001). Intuitively, each observation is weighted in proportion to its proximity to the target location. If si are discrete observations and y(si) are some function or attribute of each observation, then the interpolated value for a target location s is:

y(s)=[∑i:D(s,si)· ¿ y(si)1/(2¿2)k(D(s,si)/¿)] /[∑i:D(s,si)· ¿ 1/(2¿2)k(D(s,si)/¿)]

Rather than use smooth some spatial property of the discrete observations, si, we smooth some network statistic, z(X,i) across the spatial area. For a network statistic z(X,i) and a target location, s:

z(s)=[∑i:D(s,si)· ¿ z(X,i)1/(2¿2)k(D(s,si)/¿)] /[∑i:D(s,si)· ¿ 1/(2¿2)k(D(s,si)/¿)]

4.7.Centrality Measures for Spatially Embedded Networks

In addition to the visualization techniques, there are several new centrality measures specifically designed for spatially embedded networks. These measures are based off of the principal of propinquity described in section 2. In general, it is expected that entities will be connected to other nearby entities. This implies that it is surprising and important when distant entities are connected and when distant nodes have short paths through the network. A family of new spatial centrality measures has been developed based on that intuition, but here we focus only on a new betweenness centrality.

Betweenness centrality has been extended to a spatial betweenness centrality that places emphasis on paths between distant entities. The standard betweenness centrality for an ego node is a count of the

2 For example, observations from sensors

number of shortest paths that the ego node lies on. We define *spatial betweenness centrality* as the sum of the distances of between the nodes that the ego connects on a shortest path. In pseudo-code, the spatial betweenness centrality of a node B can be computed as follows:

```
spatialbetweenness = 0
for every pair of node, A,C:
    if B is on the shortest path between A and C:
            spatialbetweenness = spatialbetweenness + Distance(A,C)
    end
end
```

In practice, this implementation would be extremely costly in computation. Fortunately, a fast algorithm for computing standard betweenness centrality can be extended to compute this spatial betweenness centrality in O(nm) where n is the number of nodes in the network and m is the number of edges. This means that a node that lies on one shortest path connecting two nodes 1000 miles apart is equally important as a node that lies on 1000 shortest paths each connecting nodes 1 mile apart.

Trails

When observed over time, spatially embedded networks exhibit a specific kind of dynamism deserving of its own forms of analysis. Agents occupy only one location at a time, but progress from location to location longitudinally, creating a temporally embedded sequence of relationships we call a "trail". Trails are just one perspective on one part of the larger dynamic network, but thinking about relationships in sequence makes certain kinds of analysis much more intuitive. Using trails, we can begin to answer questions like, "Where do people at location X tend to go next?", "Which other agents does agent A frequently cross paths with?", or "What kind of seasonal patterns govern movements in my networks?"

Stepping back from the spatial context, we can see that sequential relationships, and the type of question we ask about them above, are not limited to tracking movement. We might also consider changes in agent affiliation, such as an Agent x Employer relationship, or changes of power, such as a Country x Political Party relationship. The formal, generalized definition of "trail" is (1) a subject node (such as an agent) and (2) a time-labeled sequence of target nodes from the same class (such as locations). We generally conduct analysis simultaneously on all the trails in a "trail set", which consists of one trail for each subjec tin a nodeset. *Any dynamic relation can be used as a trail set, so long as it has the property that at any given time, it is many-to-one (an agent can occupy only one location but a location may host many agents).* Having established that general view, we will return to discussing trails in their most intuitive context, as a description of spatial transitions. Trails and networks are closely related and defined in this way trails are actually a type of network. Although analyzing the trail as a network may not seem interesting, trails can be used to create useful networks. For example, trails can be used to create co-location or co-affiliation networks, showing who was at the same place or organization at the same time. Trails can also be used to create transition networks showing how people in aggregate tend to move from place to place or from organization to organization(2008b). Trails can also be generated from networks. Although networks do not have sufficient information to reproduce trails, networks can be used to create prototypical trails that might be expected given e.g. a transition network(Davis2008).

In ORA, trails-based analysis is conducted in Loom, an integrated tool that performs for temporally embedded trails some of the same functions OraGIS performs for spatial data. Using Loom, an analyst can visualize trail data in order to search for patterns, or export trails-derived networks such as those described above. The following section shows how to combine OraGIS, Loom and ORA into a workflow to find patterns with relational, spatial, and temporal aspects.

5. Tutorial

Now that you know a bit more about what to do, now we'll show you how to do it. We will be using the same Tanzania dataset from previous chapters and using Ora-GI to apply to methods and techniques from section 4.

5.1. Preliminaries

Begin by loading the Tanzania dataset into Ora. Now, stop and examine the MetaNetwork a bit more. Select the "location" NodeClass and switch to the Editor. Among the various attributes there you should see two spatial attributes, "latitude" and "longitude." As you might expect, these represent, respectively, the latitude and longitude of the node's location. If you look through the other NodeClasses, you'll see that none of the others have any attributes that would link them to a specific location. From now on I may refer to these Node with location attributes as simply locations.

Now, look below the NodeClasses to the Networks. You'll notice several ".... x Location" networks. In the Tanzania MetaNetwork, all of these networks ("Agent x Location," "Event x Location," and "Location x Organization") are "located-at" networks, meaning that a connection from entity A to location[3] B in one of these networks indicates that A is located at B. In this way, both attributes and networks can be used to represent spatial information.

5.2. Configuring Ora-GI

Now you know how entities in the Tanzania MetaNetwork are associated with their locations, but Ora-GI doesn't yet. Select the Tanzania MetaNetwork and open the Ora-GI tool (Menu->Visualizations->GeoSpatial Networks).

Although something is being displayed on a map, Ora-GI just guessed at how we were storing spatial information and it's probably not what we want. To make sure we know what we're seeing, we need to tell Ora-GI know how to find location information. To do this, open the "Configure Meta-Network Locations" wizard from the Ora-GI "Tools" menu. First, we need to specify which NodeClasses are locations (have location attributes). For the Tanzania dataset, this is the "location" NodeClass. Select the "location" NodeClass from the left and click the "Add GIS Attribute" button towards the bottom. Now we need to specify what kind of spatial feature the "location" NodeClass has. In addition to latitude/longitude pairs, Ora-GI also supports UTM[4] and MGRS strings. Choose "Latitude_Longitude" from the dropdown list and select the "latitude" and "longitude" attributes for their respective dropdown lists and click the "Finish" button. Now you should see a new spatial attribute listed for the "location" NodeClass. If any other NodeClasses had spatial attributes we would repeat this for those as well. Since the "location" NodeClass was the only one with spatial attributes in this MetaNetwork, click the "Next" button at the bottom.

Now we need to specify which networks should be interpreted as "located-at" networks. Similar to the previous step, first select the NodeClass on the left and then the "located-at" network on the right. Choose the "agent," "event," and "organization" NodeClasses and their respective "Agent x Location," "Event x Location," and "Location x Organization"networks. Notice that you can only select networks that go to or from one of the location NodeClasses we specified in the previous step.

Success! You should now see something like the image in figure ??? Was that a process you don't want to repeat every time you load your data into Ora-GI? Fortunately, after you successfully configure Ora-GI for your MetaNetwork, you can save that configuration as an XML file and then load it into Ora-GI instead of going through the "Configure Meta-Network Locations" wizard. To save a MetaNetwork

3 Although you are not required to put entities with location attributes in a NodeClass called location or with the type Location, it is recommended because it simplifies many common operations

4 UTM coordinates should be a single attribute with the format "<zone> <hemisphere> <easting> <northing>"

configuration, select "File"->"Current MetaNetwork Locations Configuration"->"Save to a File." To open a configuration file, choose "File"->"Current MetaNetwork Locations Configuration"->"Open From a File." You can also open a configuration file from the "Configure Meta-Network Locations" wizard.

5.3.GIS Layer Manager

Opening the Ora-GI tool should look vaguely similar to the standard network visualization tool with a few difference. First, instead of a "Legend" there is now something called the "GIS Layer Manager." This is a one-stop shop for all of you show/hide visualization needs. The Layer Manager is divided into two sections, a "Dynamic Network Layers" section and a "GIS Layers" section. In the "Dynamic Network Layers" section, you can show or hide any of the various NodeClasses or networks just as you would in the standard nework visualizer's Legend. You can also use the "Layers" menu to hide or show all of these Network layers. The bottom half, the GIS Layers can be used to show or hide any additional geographic information you wish to visualize with the network. For now, use the Layer Manager to show only the "Agent x Agent" network. If you close the Layer Manager, you can open it again through the Ora-GI "Tools" menu.

INSERT IMAGE OF LAYER MANAGER

5.4.GIS Toolbar

The Ora-GI toolbar contains much of the functionality in the standard network visualizer toolbar. The toolbar functions from left to right are: Copy map image to clipboard, Recenter, Select, Zoom-in, Zoom-out, Show/Hide labels, Show/Hide links, Show/Hide arrows, Font size, Node size(max), and Link width(max). These should be self-explanatory. Use the Zoom-in tool to select the region where the network appears on the map. Then, increase the Node Size so that you can see differences between the different locations. The default location sizes are proportional to the number of entities located there. For large networks, you may find it useful to hide both labels and links.

The Select tool allows you to see exactly which entities are located in any particular region. To use the select tool, choose it on the toolbar and then drag to select the region that you're interested in. A new window will appear with a list of all the entities located in the region you selected. To further analyze your selected region, the "File" menu will allow you to save just the selected portion of the network as a new MetaNetwork.

INSERT IMAGE OF TOOLBAR

5.5.Data Import/Export

Ora-GI can import and export data from and to a variety of different data formats, including shapefiles, kml files and image files. Data can be imported in one of two different ways: through the Ora-GI "File"->"Add Gis data ..."menu, and through the Layer Manager "Layers"->"Add Layer" menu option. In the main Ora-GI menu, simply select the type of data you wish to import and choose the file location. If using the Layer Manager, you will see a drop down of possible file formats to import. Choose the desired file format and then choose the file location.

To export data from Ora-GI use the main Ora-GI "File" menu and choose from among the "Save Map ..." options. You can export as an image (PNG), as a shapefile (SHP), as a Google Earth file (KML), or as a Ora MetaNetwork (DynetML). If you choose to save the map as a DynetML file, it will preserve any

current aggregations that you have performed. This allows you to view the co-location network at multiple levels of spatial aggregation.

5.6.Analysis Tools

The "Analyze Network" menu in Ora-GI contains most of the analysis tools you will be using. Some of these should be familiar from the standard network visualizer, especially the "Color Nodes by" and the "Size Nodes by Attribute or Measure." The color-by options work the same in Ora-GI, with the slight complication for co-located nodes. As described in section 4.3, each location is colored according to the group/component to which the majority of nodes located there belong. The size-by option will bring a dialog window similar to that of the standard network visualizer. Again, you can select either an node attribute or a network measure and size the locations accordingly. Here, Ora-GI sums the relevant values in order to determine what size to draw locations[5]. Color locations by Newman grouping and size them according to eigenvector centrality. You may need to increase "Node Size" in the Ora-GI toolbar to see differences between the locations.

INSERT IMAGE OF DESIRED MAP

5.7.Aggregation and Clustering

In sections 2.3 and 4.5, we mentioned the importance of using multiple levels of aggregation in order to reduce the risk of committing the ecological fallacy. In Ora-GI, the Network Aggregator is the tool to use. You can find the Network Aggregator in the Ora-GI "Tools" menu. To use the Network Aggregator, simply slide the bar on the top to the left or right. When you let go, Ora-GI will re-aggregate your MetaNetwork and display the results. Below that slider tool, you'll see two large bars. The green bars represent the proportion of the network information (left) and spatial information (right) that is preserved, while the red bars represent the proportion that has been lost by the current aggregation.

INSERT SCREENSHOT HERE

5.8.Ora-GI Tips & Tricks

Ora-GI has some time-saving features that can make it easier to get your work done and also make it easier to reproduce it later. One of the tools that can help with this is "Get/Set View" tool in the Ora-GI "Tools" menu. This lets you more precisely set the viewable map region. By copying and pasting all of the numbers you see there, you can be sure to get matching map images even if you close out of Ora-GI or accidentally zoom in somewhere. In general, whenever you see a number in Ora-GI, it's there to help you document your process and repeat it.

5 See section 4.3 for more information.

If you find yourself loading your own geographical data into Ora-GI, you may want to change the default background data. You can do this at any time through the "File" → "Current GIS Layers" → … menu options. In addition to changing the default background data, you can create several different layer configurations that you use with different datasets. To do this, use the Open From/Save to File options instead of changing the defaults.

Sometimes, Ora-GI may get just a step or two behind in what you're trying to do. If you hide/show some networks, size by a network measure, etc. and the map doesn't seem to have updated, try refreshing the map by selecting the Recenter tool and clicking on it.

5.9.3D Visualization and Kernel-Smoothing

In order to use the smoothing methods discussed in section 4.6, we need to use Ora-GI's 3D spatial visualization tool. In addition to the 2D visualization tool you've used so far, Ora-GI also has a 3D visualization tool built using the NASA WorldWind Java application. Before opening the 3D visualization, check to make sure that the drivers for you computer's graphics card are up-to-date. The 3D visualization is much more demanding and requires modest hardware accelerated graphics. If, after switching to the 3D visualization, you only see a black sky, your computer may not be able to use the 3D visualization.

To open the 3D visualization, under the Ora-GI "Options" menu, choose "Use 3D Visualization." This will switch from the 2D visualization to the 3D visualization, so take save any images or other data files before doing this. Once you open the 3D visualization, you should see the globe floating off in the distance. The 3D visualization has different mechanism for manipulating your view from the 2D visualization. Rather than clicking to recenter, you can either click and drag the map where you want it or navigate using the arrow keys. To zoom in and out you can use either a mouse scroll wheel or the "+" (no shift) and "-" keys. You will also see a new menu, "3D Options," containing several options. With the 3D visualization, you can choose between exploring a 3D globe, and a flat projection of the globe. The other new option, "Show Heatmap Options" lets you use the smoothing methodology described in section 4.6.

There are three different ways to use the smoothing methods: Node Intensity, Node Attribute, and Node Measure. The Node Intensity option will show a heatmap with the interpolated density of entities in the MetaNetwork shown on the map. Selecting Node Attribute or Node Measure will use the kernel smoothing method to interpolate either the expected node attribute or the expected network measure for unknown nodes at every point in space. All of these options require specifying two parameters, a bandwidth parameter and a color sensitivity parameter. The bandwidth parameter indicates how much to smooth the values. Higher values will lead to more smoothing; lower values lead to lower smoothing. The color sensitivity parameter indicates how much to skew the colors in one direction or another. Higher values skew colors towards red with lower values skewing them towards blue. Both of these parameters may need manipulation to yield a helpful image. For the Tanzania MetaNetwork, choose Node Measure with eigenvector centrality and use a bandwidth of 20 and a color sensitivity of 0.212.

6. Geospatial Assessment Report and Spatial Betweenness

In addition to the Ora-GI visualization tool, Ora has a Geospatial Assessment Report that is designed to yield useful information about spatially embedded networks. After choosing the Geospatial Assessment Report, you will be asked to specify a location NodeClass (must have latitude and longitude attributes) as well as a X by location network to analyze. Among other things, the Geospatial Assessment Report will return the spatial betweenness values computed for all X by X networks in the selected MetaNetwork. For the Tanzania dataset, select the "Agent x Location" network and view the resulting report.

INSERT SAMPLE REPORT HERE

7. Problem Sets

1) Load the Tanzania MetaNetwork into Ora-GI and configure it using the tanzania_config.xml file.
   (a) Compare betweenness and closeness centralities in the social network.
   (b) Now, compare betweenness and closeness centralities in the entire meta-network.
   (c) What differences do you notice? Can you explain these by looking at the meta-network in the standard visualizer?
2) Still using the Tanzania dataset, what happens if you restrict the social network to nearby connections?
   (a) Use the Geospatial Assessment Report to create a new network that links people within a moderate distance, say 500 miles.
   (b) Perform an AND with this network and the original social network
   (c) Compare the merged network with the original. How did the density change? Would this be evidence for or against propinquity in this network?
   (d) Rerun the Geospatial Assessment Report on the new meta-network. How did the list of high spatial betweenness centrality people change? Which list do you think is more useful/accurate? Why?
3) Load the drug_gangs.xml file into Ora. Select the last time period and load it into Ora-GI. Configure it with the drug_gangs_config.xml file
   (a) Perform a cohesive grouping analysis. How are the groups distributed spatially?
   (b) Run the Geospatial Assessment on the "seizureAttendance" network. Which agents were most important?
   (c) Create a smaller meta-network with only the important agents. Where were these important agents located?
   (d) What does it mean for an agent to have high spatial betweenness in this dataset? What role might these people have played?
4) Ora-GI helps you find spatial patterns in network data. Specifically, by coloring by subgroup, or sizing by a measure, .it lets you look for spatial patterns in the structure of the network. For which kind of networks is this most useful? When might this not be useful?

CHAPTER 7:Trails

Now we explore trails. Trails are paths that Whos move through within a network. Naturally, trails involve both a Who and Where and even a When. When you can ties those entities together over period you have a trail.

Julius Caesar has seen his empire and bureaucracy ever expanding. In an effort to better analyze the structure of his empire, Caesar decides that he wants to know exactly how his military command travels the known world.

We already employed extensive tools in the analysis carried out in the previous chapter using bipartite data and even data that makes use of three entity classes. However, what can aid us in discovering how the command staff in Julius Caesar's empire travels the known world?

What can we learn from carrying out such an analysis? Will interesting structure evolve that may be of importance to the Julius Caesar? Are there intricate patterns unobservable to the computer unassisted in terms of where such military personnel travel? What if it is found that some of his key advisors often find themselves in the same locations as some of his key political adversaries? What then are to make of such location similarities?

Terms to understand

Trail: A set of nodes and links that form a single unbroken chain, such that no node or link is repeated.

Path: A set of nodes and links that form a single unbroken chain, such that no nodes and links can be repeated.

Could it be possible we could extract even more useful structure that could again reveal to the senior military command in Julius Caesar's empire, how the company really functions versus how they believe it functions?

Could the emerging travel patterns lead Caesar and his staff to draw new conclusions about where the business is located or where it should open new locations to better serve the geographic needs of the empire? Could we accurately predict the channels that are vital to the continuing success of Julius Caesar's empire? After all, how does Julius Caesar know that his extensive group of mercenary soldiers travel and where they go? How important is it for military analysts to understand the travel patterns of either an individual or group of enemies. Is there something imperative to be gleaned from how certain enemies were trained and how the very same terrorists took divergent paths and then a few years later, they meet up to form a battalion, in say, the outskirts of Rome? It would behoove Caesar to know about trails then.

What about all the points they shared in common, such as cities they passed through, houses they stayed at in various cities, routes they took, stops they made, on their way to forming the enemy army? Are there tools for Caesar's military analysts to employ to look at such patterns and pull information from them in such regards? What could such patterns reveal to the analyst? Could the analysts see a network structure where it otherwise might be impossible to see without the aid of the computer and sophisticated algorithms?

Let us revisit Caesar's administrative advisor from the previous chapters. His job is to see that his network is running as smoothly as possible. Is there anything to be learned about a network of servants and how they travel throughout Rome? In other words, when Caesar requests a task to be completed

what trail does the task carrier flow to get the requested information to Caesar? If studying this information, could some meaningful network data be extracted that might be of interest to the dynamic network analyst?

What if it could be ascertained that information traveled along informal networks in certain patterns to certain persons in Caesar's government? Could the government then be optimized based on such network information flow? What would be the best way to do optimize it? In all cases, Caesar's analysts would need a tool that would enable him or her to extract this data. What tool would that be? As we mentioned before, in the parlance of dynamic network analysis, we would say he is interested in learning about "trails."

What is a trail?

Just hearing the word trial conjures up an image of a path in the woods; the beaten down part much traversed and walked over through the years that leads one through the woods. It would take someone that enters the beginning point of the trial all the way to a certain destination at the end of the trail. From afar, we only see the woods.

The dense, thick, foliage and that is about it. This is one case where we see the forest for the trees. A trail might conjure up images of Caesar's soldiers traversing the yet to be conquered lands, using the existing trails of the people there to guide one through the woods to safety. Perhaps, naturally, we would guess that the trails make sense on some levels. That is they probably avoid parts of the geography that are difficult to traverse. Most Indian trails and animal trails for that matter meander their way logically across geographically diverse terrain in a way that is conducive to migration. Many human trails simply mimicked animal trails in that regards.

You wouldn't blaze a trail through the mountains by climbing all the tallest peaks you found would you?

You would take the path through impenetrable swampland and quick sand, even if it was the shortest geometrical path to get from point A to Point B. Most likely, you might take the path of least resistance.

You would take the path around the mountains that is easy to walk. You would avoid the quick sand and take the land that is more stable and sturdy. You would walk "around" tall mountains not up them. You would take calm rivers, not raging rapids. So what does this all say about trails? And likewise, how would Caesar best march his armies on long runs to sack enemy cities? But let us look at the bigger picture for a second.

It essentially says there is some logical reason a trail evolves. They are not random. There are many factors that affect how a trail would form. It is no different with network trails in the context of dynamic network analysis. It might be said in the context of dynamic network analysis that a trail is the tracking of an individual actor or multiple actors over time through the "woods" of any complex multi-agent multi-modal network. The actor can literally be any entity but in most cases from the standpoint of the dynamic network analyst would constitute a person such as an enemy or real person. In more technically speaking jargon, a dynamic network analyst would consider a trail to be the "observable portion" of a system which includes hidden context variables of network structure. Those variables might include "activities" or "goals" motivating an agent to do certain things or carry out certain tasks. Such a model of trails can be used to make "inferences" about hidden states of any actor inside the network.

As such a network model can be developed using observations about actual movements through a network. But there are a lot of factors a dynamic network analyst would use in constructing a trail based on observed data. These variables would include making observational statements about an agent or actor's behavior. Dynamic Network Analysts have come to identify that many actors in a network will share similar behaviors and that those similar behaviors can be used to construct trails or address the likelihood that any actor would follow a similar path through a network.

Trails and geography

We can see from our natural understanding of what a trail is that a trail must, in some form or another, be part of a network path. That path typically in real world applications usually involves geography. In Caesar's world, this would constitute the Mediterranean and all the surrounding landmasses and yet to be conquered lands.

Although it can be said that you don't need geography to perform network analysis but just think how important it is when considering any element of a network. It is like the whole theory of Relativity: time and space are inextricable from one another.

Philosophical opinions might be considered in the geography of Philosophical inquiry.

An enemy of Caesar's doesn't operate in a vacuum. He soldiers in operations across many continents or within the narrow limited confines of a sleeper cell, waiting to be activated by an outside agent.

Here we will depart from our Caesar model and consider a real world application of applying trails to network analysis.

Once such application of identifying trails to use to reveal network structure was conducting by Carnegie Mellon University's CASOS lab. In a paper co-authored with PhD. Student George Davis, we were tasked with developing new computational techniques for Merchant Marine behavior under a Social Network Analysis framework. In this experiment doing we were presented with geospatial data from AIS transponders obtained from over 1,700 ships traversing the English Channel over a 5-day time period. The analysis performed contained three phases:

1. Spatial clustering algorithms were used to detect places of interest and relationships among ships in the collected English Channel data.
2. An extraction of relational information, which was analyzed in network form. A suite of network analytic measures were applied to locate patterns in the network and determine at what individual node level those patterns were discovered.
3. We applied an intervention analysis which models an intervention (surveying ships at ports) and suggests a strategy for allocating surveillance.

This purpose of the English Channel analysis had two primary goals: Firstly, a rendering of as much information as possible regarding merchant marine networks and behavioral patterns on the basis of the data given. The patterns detected should inform future research efforts to better understand the community; secondly, obtain an assessment of the tools and techniques applied as potential parts of an analysis regime which should be repeated on data gathered in the future.

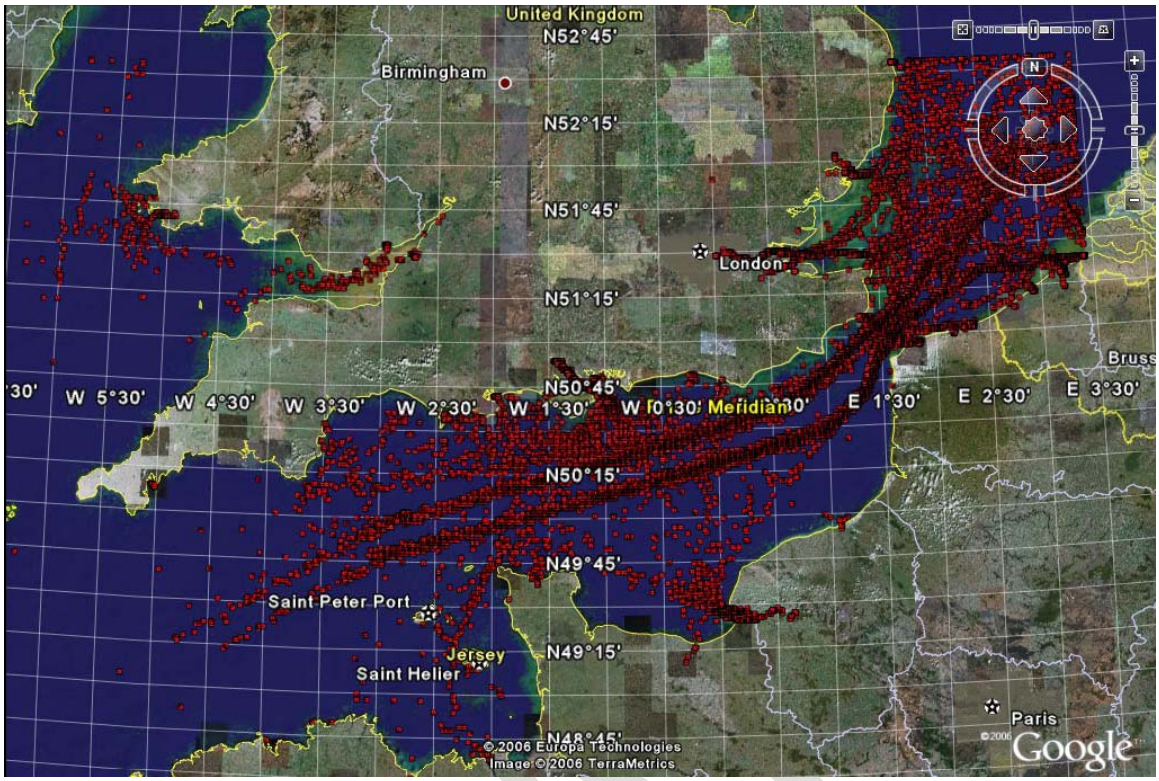


Figure 29: The English Channel Merchant Marine network

In the experiment conducted by CASOS, information was obtained from using GPS data. This information was used to analyze the position of ships traveling the English Channel over a 5 day period. Figure 1 shows the shops at a relative position on a particular day at a particular point in time. The ships positions were then plotted on a map of the English Channel to obtain the visualization above.

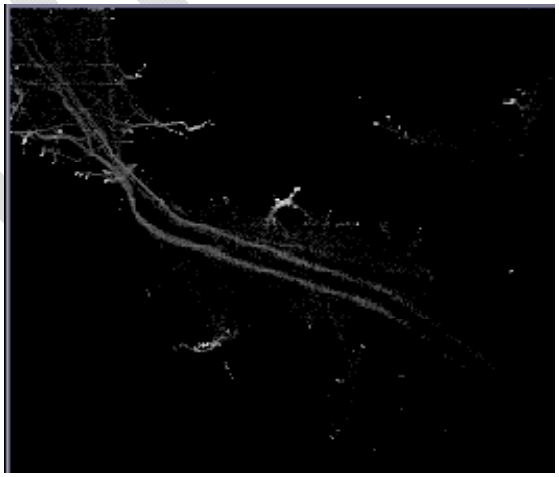


Figure 30: Observational Density

In Figure 2 we remove the geographic map and gain a visualization of the shipping network in the English Channel. We can see that there are two primary shipping lanes that appear to stretch across the model of the network.

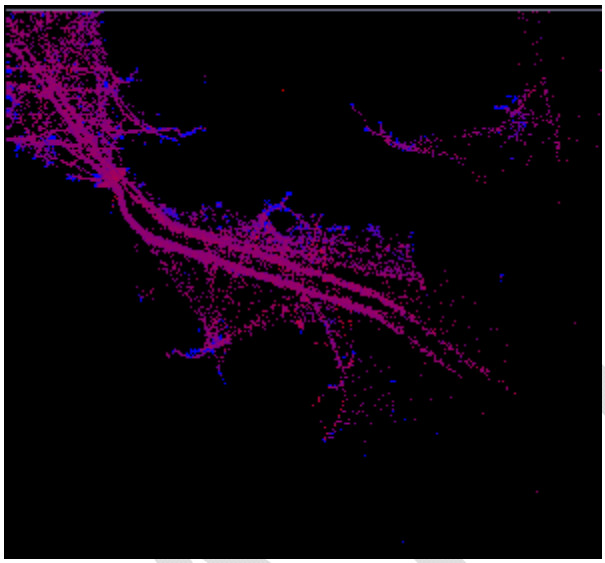


Figure 31: Average Speed

In figure 3 we have a visualization of a network obtained by analyzing the relative speed of the merchant marine vessels across the 5 day period. We see thicker, that is weighted ties, where the speed is the greatest.

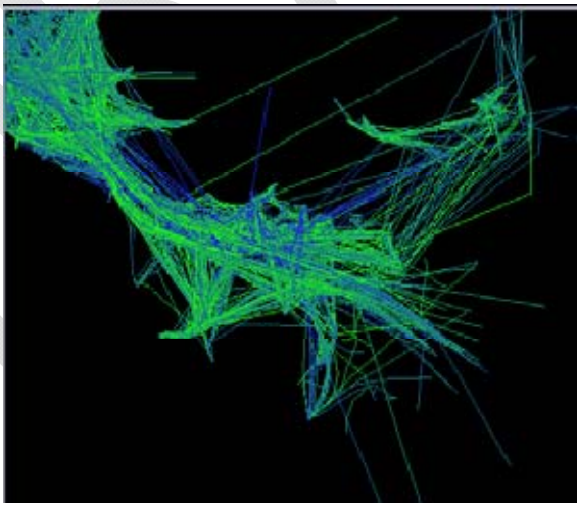


Figure 32: Ship Paths

Finally, in Figure 4 we have a visualization of the overall network contained both the path of the ships and the trails from which they follow. Now what does this visualization tell us? The visualization tells us many important things.

The most import aspect of this experiment reveals a complex multimode structure evident in the English Channel. We can see network trails that are of primary importance in the strength of this network. We know from previous chapters that once we know what a network looks like, we can begin to make highly probably observations and inferences about what keeps a network strong and what it might be vulnerable too. For instance, in the above example what would happen if one of the main shipping arteries was incapacitated for some reason? What would be the likely result and how would the network respond?

Those revelations were obtained by creating a network by following actors or agents through periods of time and then plotting the relational position on a geographical map. Seemingly by following the trials of ships were discovered a picture of the network shipping map that comprises the shipping lanes of the English Channel. How important might that be to network analyst?

Well imagine if we are given any random ship that is entering the English Channel and the authorities are alerted that such a ship might be carrying contraband or other could be involved in a blot to sneak in a bomb into a harbor. How might the authorities immediately go about isolating that vessel? Would a map of the network the boat would likely follow come in handy to the analyst attempting to figure out a way to avert a crisis? Surely it would.

Trails are everywhere

We should see by now that that it is highly important to consider the trails in the context of where they are located. He would clearly gain an advantage if he knew all the critical shipping lanes in the Mediterranean ocean.

Trails can be powerful network representations of data when considered in the geographic context. Below are a series of images which further illustrate how the dynamic network analysis plots a relational network data on a geographical location.

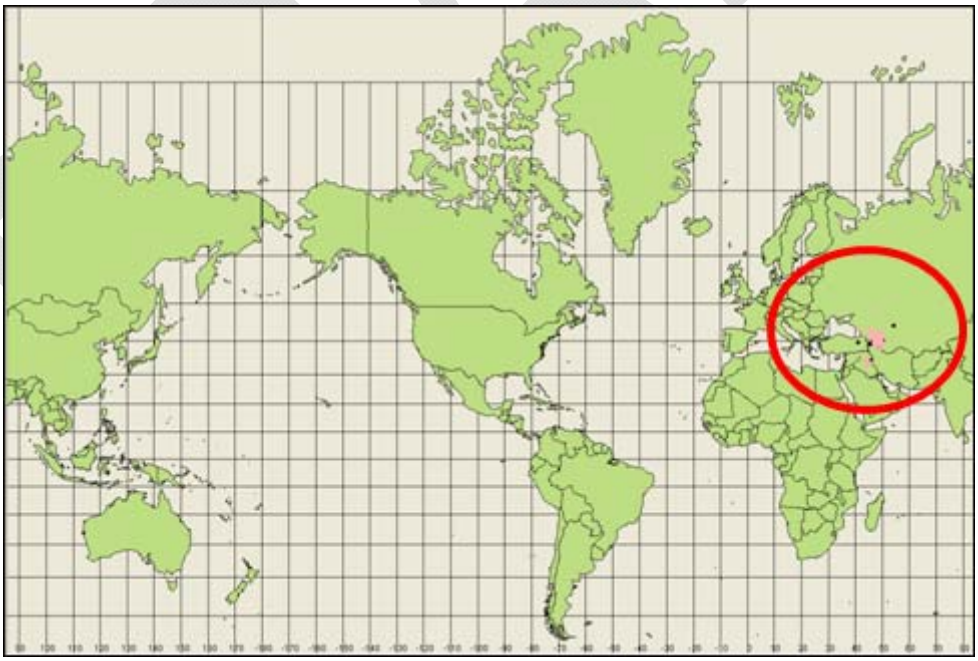


Figure 33: World Map Example

Figure 5 shows that we can plot nearly any data on a world map and learn important information about a network simply based on where that network is located. In this example we have localized information relating to a network that is centered in the area of south western Russian.

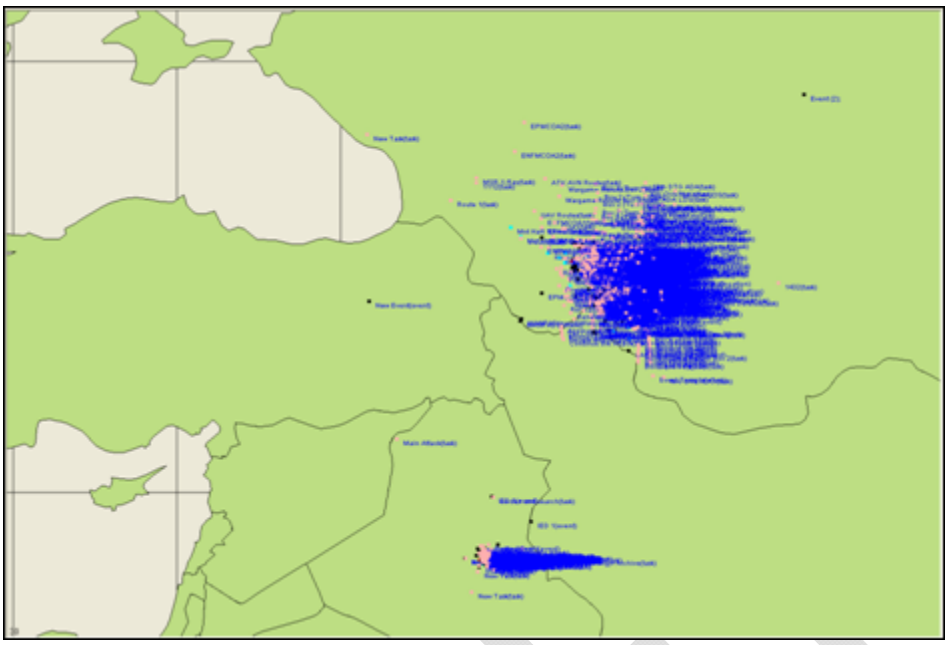


Figure 34: Network Cluster on Map

In figure 6 we can see the result of plotting clustering on a map.

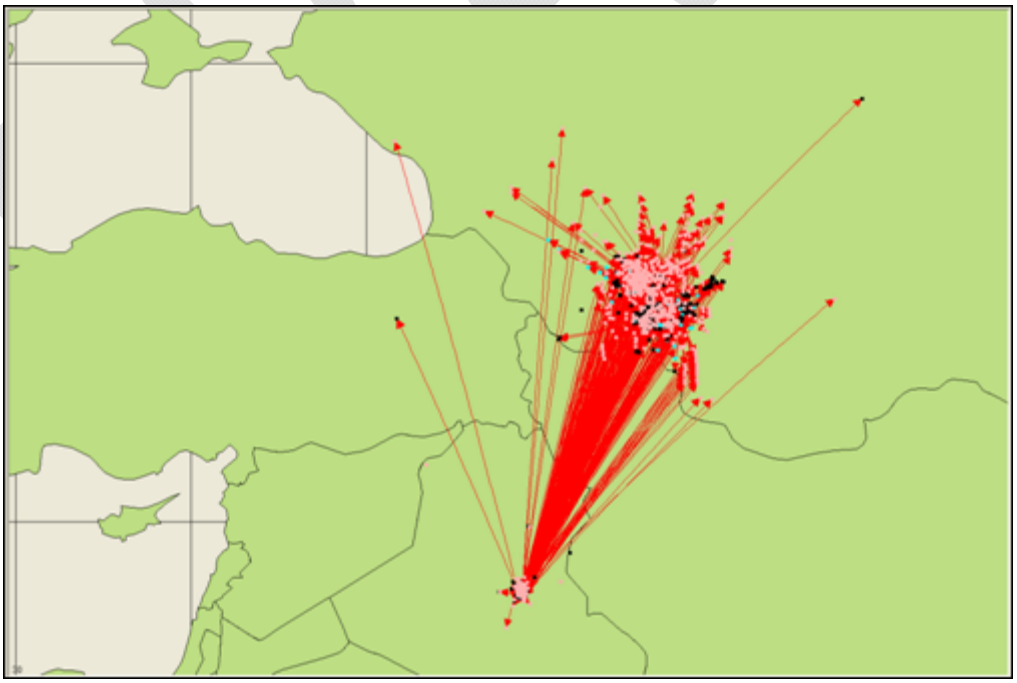


Figure 35: Network over geography

By connecting the dots, we can grasp visualizes some powerful insights about a network, which is by definitions multi-modal as it contains both actors and locations.

Loom

Another aspect of network trails in called "loom." Looming is an effort to study the patterns or paths of actors over time and determining if for some reason that actor seems to gravitate toward a certain location and what exact paths an agent, actor or who may have taken to get from one point to another. Why is this important? Well let us revisit our hypothetical examples at the beginning of the chapter.

Does it make sense for Military advisor to know what points across the geographic Mediterranean that their soldiers seem to keep revisiting and would such knowledge prove advantageous to the general marching on orders from Pompey or Caesar or Antony or Brutus? Would it be helpful to understand what waypoints seem to be of particular importance in terms of resources for the army? What else could be gleaned from seeing which locations are often revisited the most?

It probably seems even more evident when you consider "looming" in terms of Caesar's military analyst interested in tracking particular army's movements. Where does this army seem to go most often? Why would such an enemy appear in a place such as North Africa with regularity? What other cities would the enemy have a reason to visit?

In terms of the computer network analysts, why are certain servers constantly accessed and others are not? Could it have something to do with the power of the server? Could the very fact that one server in particular is accessed more than others clue the analyst in that there is something uniquely interesting about one server that makes it stand out more so than the rest? There is probably a good chance that this is indeed the case.

We can see there is much to be learned from ascertaining trails from network structure, but there is more information that is needed to gain a clear understanding of what a trail is and to ensure that that trail is meaningful at least in the sense of sort we want to consider analytically. How do to this is create what is called a trail set.

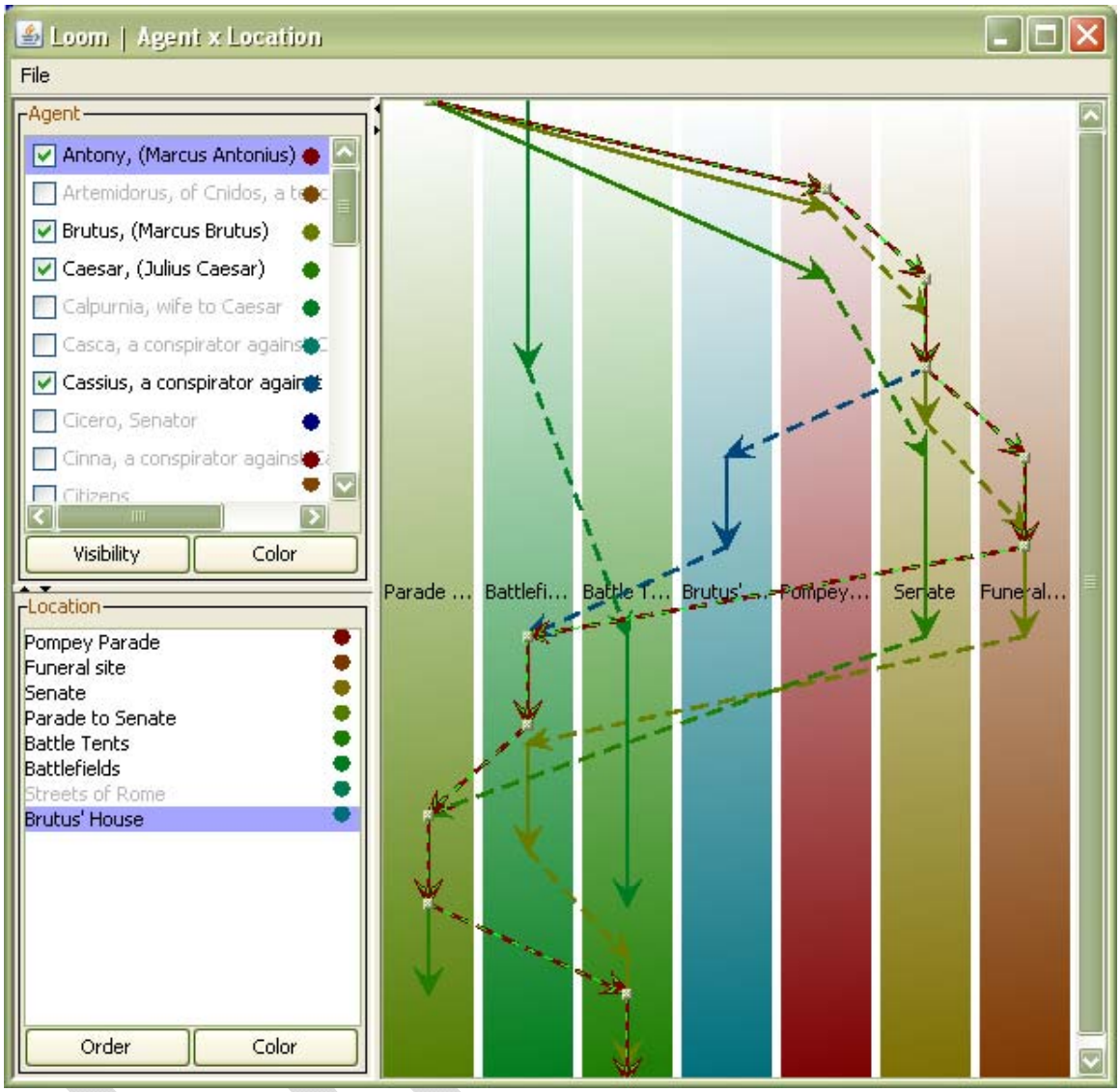


Figure 36: Loom visual of major adversaries x location in Julius Caesar Meta-Network

In figure 29, we created a trail set in ORA, but inputting locations from the Julius Caesar Meta-Network and graphing them as columns against a charting function of Agents. We could correlate all agents but that is probably more detail than we need. By plotting the five major adversaries on the Loom tool, we can follow their travel paths from one location to another.

This visual tool is very useful in understanding the importance of waypoints. We can see from our Loom example that many of our adversaries converged on the parade and on the battlefield. How sad, from good times to decidedly destruction. It must also be further noted that the arrows are imputed travel paths based reflecting in the appearance of a node at a likely location. In other words, we can infer a stop at point B if our agent left point A and arrived at point C an d the only way to travel from point A to point C if by of B. We can inform then they stopped at point B(Davis 2007).

Who, What, When, Where, and How

We all remember the qualities of a good composition from grade school. In case you forgot, let us take a quick refresher course. With any type of composition or story specially one of news nature, we are concerned with several key piece of information: *Who, What, When, Where, and How.* This is something we have been dealing with for several chapters now.

Such factors are all the critical aspects of helping reveal a network vis-à-vis a trail and to do so we must consider properly the weight of such aspects.

We are reminded once again of the plot to kill Caesar. In our scenario, we are told that someone has been killed but let us pretend for a second that we don't know who was really behind the plot. We will put ourselves in the shoes, eh, sandals of Marc Antony and pretend he was clueless as to why anyone, let alone Brutus, would wan to murder Caesar, his beloved friend. Marc Antony would now start constructing a trail set of *Who, What, When, Where and How* to see which person had a unique path to the murder of Julius Caesar.

Let us also say Cassius was in the senate chamber at the time the murder has happened, which was correct. The "Senate chamber" which you are probably guessing correctly by now is a "location." We need to ascertain such important factors as motivation for killing Caesar in Senate and learn if Cassius would be our chief suspect based on motivation. What about is role? What about his knowledge? His resources? Are you beginning to see the importance of trails? We are sure Marc Antony is.

Did Cassius have access to a knife? Does he know how to hold a knife? What about his knowledge? What if it were proven that the man killed had a wife who was involved with Caesar and this person was entitled to quite an inheritance if something dreadful should befall the man in the kitchen, which is exactly what happened.

So we want to solve this crime. How are we going to solve the murder? We will first look at the network that both Cassius and Caesar were part of. This will underscore the power of trails and who they fit in the context of networks. After all, we don't want Caesar's murder to go unsolved – okay, we are taking literary license with this. Just play along. How could we ever life with ourselves?

So, we are about to consider the actors involved that might have possibly killed Caesar. Motives, alibis, availability all depend on connections and are all factors that are highly critical in solving the murder. There are others. First, let us ask these questions: was it Caesar in the Senate with the knife?

If we can prove that, he may emerge to become our chief suspect. To prove this assertion, we resort to network analysis using trails. We will construct a tail by obtaining other facts that tell us something about the network Cassius was involved in. We would proceed in a manner as such (Davis 2007).

- No – It couldn't have been Brutus. Cassius gives him an alibi
- No – Mettelus Cimber was in the antechamber next door and we know no crime was committed there
- Maybe it was – neither. Do we have evidence to mutually rule them out?

So what have learned? It should be this: links among the *who, what, where, how, why and when* are then critical to establishing trails.

By having a bit of an understanding about the motivation and the stories among those factors helps ups connect a network that may have existed over time that could have resulted tin the murder of the Caesar in the Senate. In doing so we should be able to exonerate or implicate Cassiuss.

We would have considered what his knowledge was and what his motivation for being in the Senate could have been. What are his beliefs about Julius Caesar? What did he know about him and what of his relationship with someone that was directly connected to Caesar?

How all these facts would be taken into consideration when it comes time to discover why the man in the kitchen was killed and for what reasons?

You see what we are doing here is really creating a network of everyone that is in any way shape or form connected to Caesar's world. It is that world of networks in which Colonel Mustard's true guilt or innocence may lie. So it is critical to understand all of those factors... Without them, we really don't know much. With them, we may know all we need to know.

From there, having gathering all such evidence, the dynamic network analysis could plot out over a map and timeline to discover exactly what happened or what likely happened. They DNA analysis would know who what where and when. Who had the most going and how behaviors we are modified or changed over a time period based on certain events? So what does this tell us?

It tells us that the dynamic network is inextricably connected to many variables, which need to be studied in great detail and analyzed. Those connections can be made. We can learn an awful lot about who was likely to kill Julius Caesar. Facts will begin to emerge that can really determine who killed Caesar in the kitchen and if it was most likely Cassius or somebody that wanted to set Cassius up. Conversely, it might vindicate him or seal his fate as the guilty party.

CHAPTER 8: Evolving Networks

We should know by now that networks evolve and change over time and it is the key role of the dynamic network analyst to observer those changes and predict how networks evolve and change over time. In fact, it is this time consideration that marks the true difference between dynamic network analysis and traditional link analysis.

When we revisit ABC Corporation imagine how helpful it will be to have a way to predict how a network will evolve given a certain changed variable, such as the removal of a "key" manager. Wouldn't it be helpful to know how the isolation of someone within ABC Corporation would impact the efficiency of the overall network?

Let us think of the CIA analyst as well. Here we can see how critical it is to identify a particular terrorist and ascertain what impact to the overall network would result if such a terrorist was removed from the network. In fact, using a data set from the Tanzania Bombing in Dar Es Salam in 1996, we will demonstrate different experiments involving time and network evolution. We will see what happens to the overall structure of the network once key agents within it have been isolated or removed. How will the dynamic network analyst perform such an experiment? Once again, we are taking back to our measures – that is algorithms – that perform powerful calculation on the network data describing what is most likely to happen as a result of one key agent's removal or isolation.

What about our network server administrator. Perhaps it would be of key importance to such a person to understand how the removal of one key server is likely to change the existing server infrastructure. Therefore, it should be obvious, what benefit this would have to such an IT professional.

Near Term Analysis

In this step, we use a Tanzania bombing dataset to see how Near Term Analysis (NTA) in ORA is working. NTA is an analysis function utilizing a multi-agent model, Dynet.

Particularly, NTA assumes a set of node removals from a given organizational structure and estimates the performance changes and the emergent structures. Thus, the *what-if* analysis with NTA will provide an answer on how the organization will behave and change with a sequence of strategic interventions or unexpected personnel loss.

This demonstration hypothesizes two agent removals: Wadih-Al Hage and Ahmed the German, from the bombing dataset.

1) Loading an organizational structure and starting the Near Term Analysis function

The target data should be loaded in the dataset management panel in ORA main window, as in Figure 1. Also, the data must be highlighted before starting NTA.

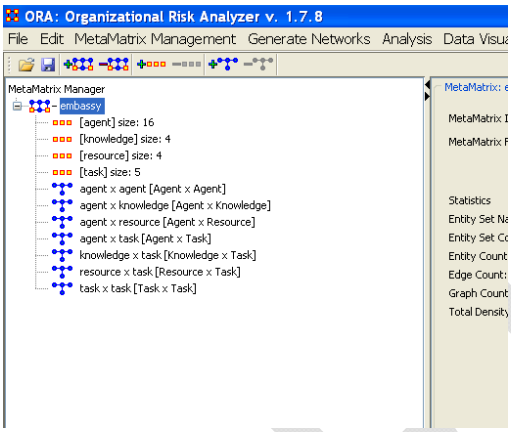


Figure 37 a loaded dataset in the ORA main window

Under the Analysis menu in the ORA main window (Figure 2), you can find 'Near-Term Analysis.' Click the menu to start NTA.

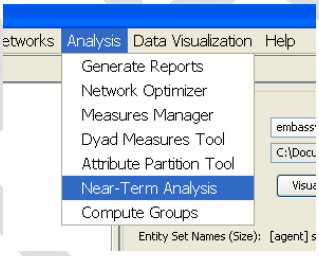


Figure 38: menu for starting Near Term Analysis

If your dataset has all the necessary information for the analysis, ORA should display the NTA main window, Figure 3.

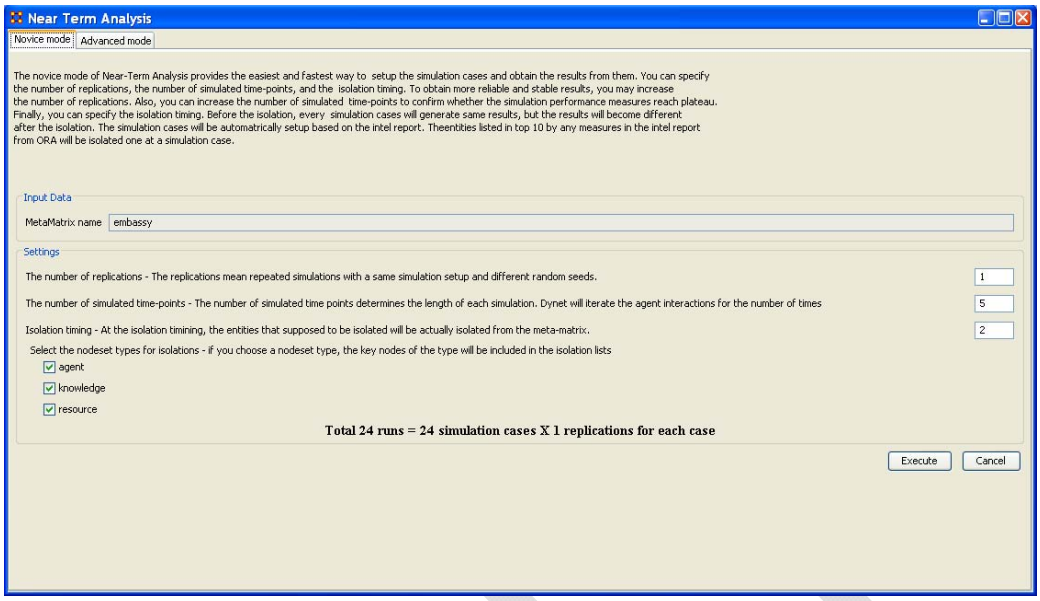


Figure 39 a novice mode of Near Term Analysis

NTA has two modes: Novice and Advanced (Figure 4). Novice mode provides an instant analysis setup, so you can just click the button, 'Execute', at the bottom of the window. However, this demonstration will not use this mode because Novice mode does not allow users to setup their own analysis question and virtual experiment hypothesis. Instead of using Novice mode, we will use Advanced mode, and you can switch Novice mode to Advanced mode by clicking on the tab of Advanced mode.

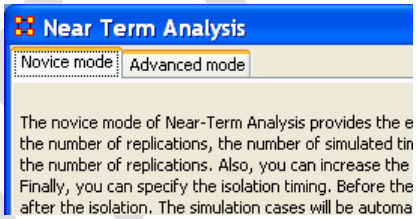


Figure 40 tabs for Novice mode and advanced mode

2) Analysis Setup in the Advanced mode of Near Term Analysis

After clicking the advanced mode tab, the NTA main window will change like Figure 5. We will proceed with the rest of the demonstration in this mode. The NTA main window consisted of four sub panels: input panels, settings, simulation time line and cases to simulate. We will see the usage of the sub panels throughout this sub section.

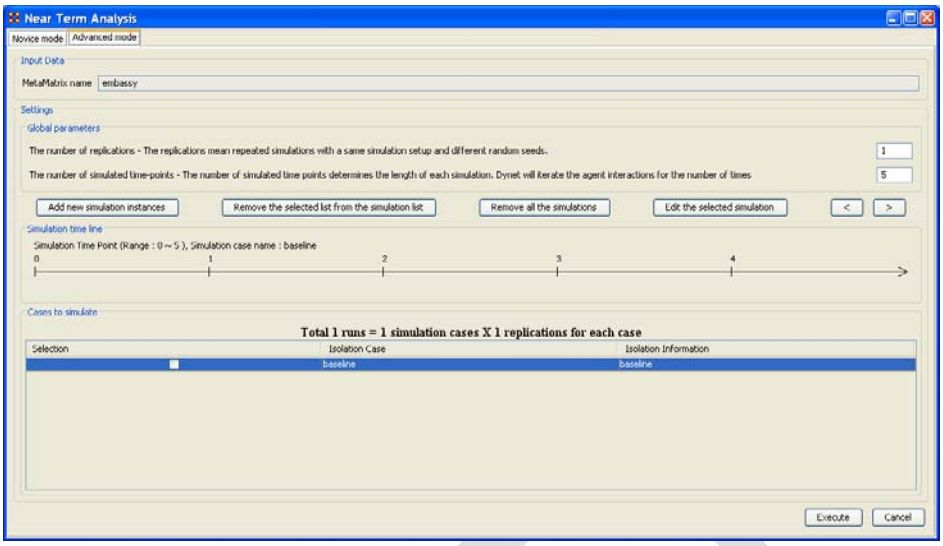


Figure 41: an advanced mode of Near Term Analysis

First, we change the global parameter in Settings. As in Figure 6, we change the number of replication and the number of simulated time-points to 2 and 52, respectively. The inside analysis engine of NTA is Dynet, a stochastic multi-agent social simulation. Therefore, we specify the number of replication and the simulation length for each virtual experiment cell. After specifying the simulation length as 52, you should see that the simulation time line expands to 52 time steps.

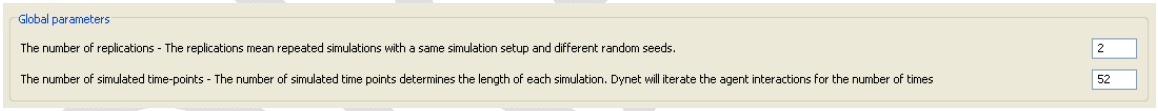


Figure 42: global parameter setting

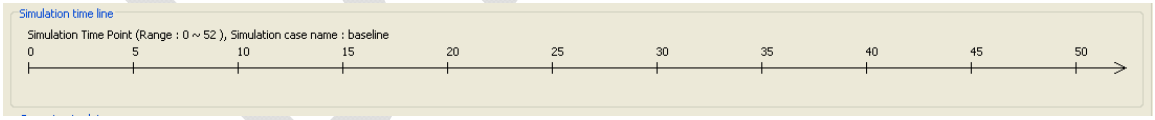


Figure 43: Simulation time line

After setting up the global parameters, we setup the virtual experiment cells. Initially, there is a pre-defined simulation cell, Baseline. Baseline is the cell without any node removals, so it can be a result when the organization does not experience any interventions. Besides Baseline, we will setup two more cells, the removal of Wadih-Al Hage and that of Ahmed the German. We start setting an experiment cell by clicking "Add new simulation instances", located above the simulation time line and under the global parameters.

After clicking Add new simulation instances button, you should see a dialog, titled as "Near Term Analsys – Simulation scenario create wizard." This wizard supports the creation of virtual experiment cells in three different ways. First, you can setup as you want by choosing the first option, and we will use this option. The second and the third option will ask you a set of criteria for selecting important agents in the network and make experiment cells according to the selection. Therefore, the first option give full flexibility to users, and the second and the third option provides a systematic analysis setup method. To proceed this demonstration, we select the first option and click "Next".

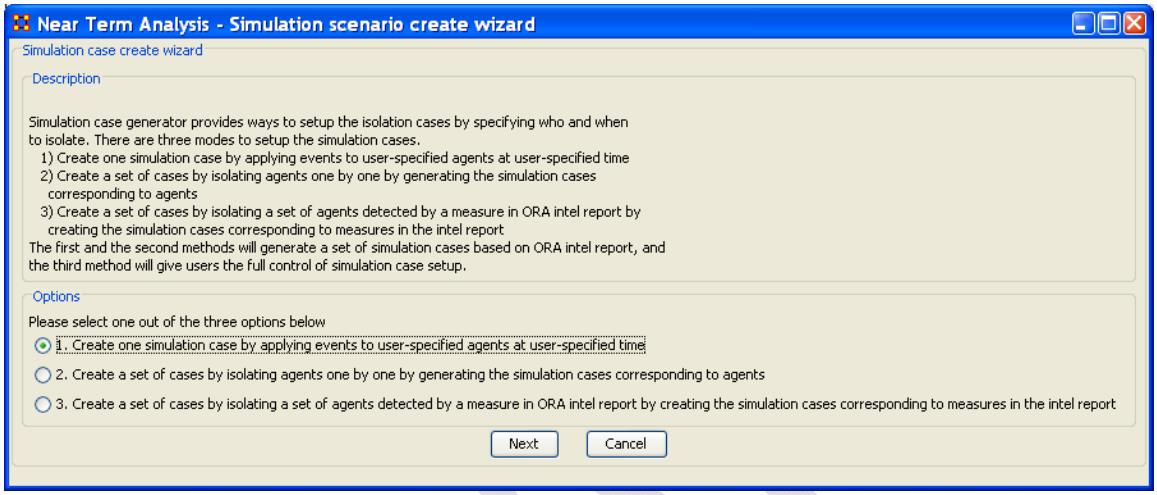


Figure 44: a simulation scenario creation wizard

Selecting the first option brings the dialog window in Figure 9. We can setup events by using the table in the window, add the events to the simulation case, and add the finished simulation case to the NTA main window.

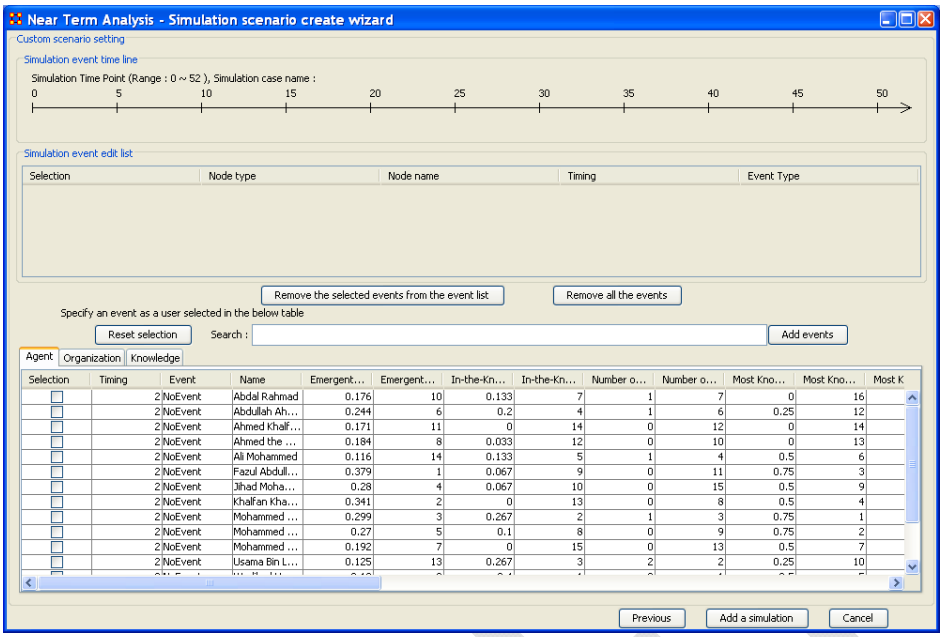


Figure 45: a scenario creation wizard for 'user specified event option'

Fist, we will make a case experimenting the removal of Wadih-Al Hage. This case consists of an event, the removal (isolation) or Wadih-Al Hage. To create the event, we find his name in the table located at the bottom of the dialog window. Then, we check his name in the left most check box, and change the event timing at 10 (or the time step when you want to isolate him). After that, the setup should look like Figure XXX.

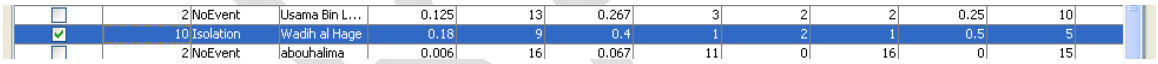
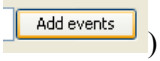
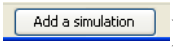


Figure 46: select an agent and assign an event and its timing to that agent

We finished the setup of one event, so we have to add the event to the simulation case. To add the setup event, we click 'Add events' button () located next to the search text input display. After adding the event, the time line in the dialog will change as Figure 11, which means that we are isolating one agent at time 10. For now, we finished making one simulation case because we are removing just one agent for each case. To finish the setup of the case, we click 'Add a simulation' button located at the bottom of the dialog (). After clicking 'add a simulation' button, the dialog will disappear, and the NTA dialog will have one simulation case at 'Cases to simulate' display.

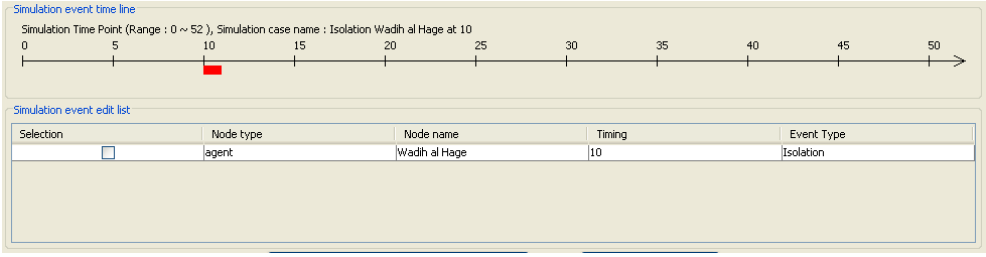


Figure 47: after 'add event' in the case creation wizard

Just going through the above one more time, we create one more simulation case by only differentiating the removed agent name, from 'Wadih-Al Hage' to 'Ahmed the German'. Thus, we have two removal simulation cases in the list.

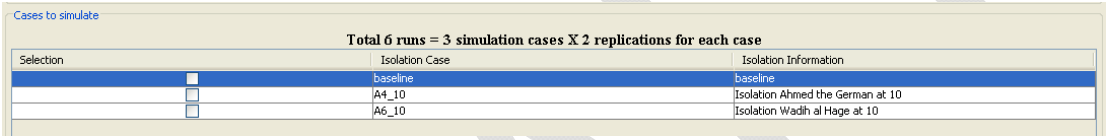


Figure 48: simulation case list in NTA main window after 'add simulation' in the wizard

NTA dialog window tells you that we have total six simulations to run because we have three cases (one baseline + two removal case) that should be replicated for two times. Also, you can highlight the simulation case by clicking it, and the simulation time line will change as you change the highlight. The simulation time line tells how the simulation case is designed.

Figure 49: information on how many simulation runs will be initiated

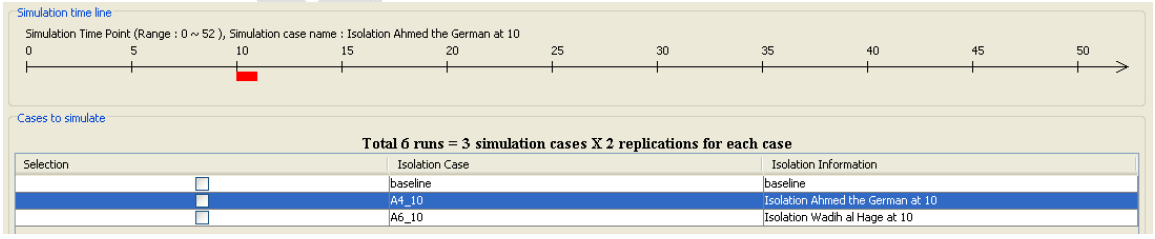


Figure 50: simulation time line display for each of different cases

3) Running the Near Term Analysis

Since we finished setting the three simulation cases, we have to execute the simulations. We click 'Execute' button (Execute) at the bottom to start the simulations. After starting the execution, you will see a small warning dialog, Figure 15. This is just a warning about the long execution time when you have a large network. This dataset should not take such a long time. After clicking 'OK', a dialog will display how many simulations are done, Figure 16.

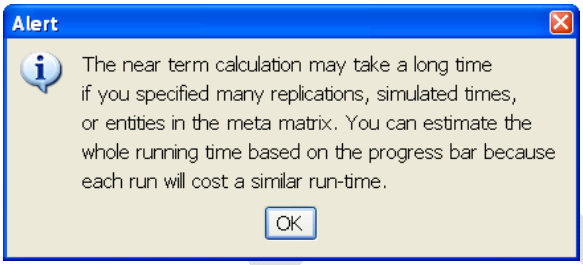


Figure 51: a warning about long simulation time when the dataset is big

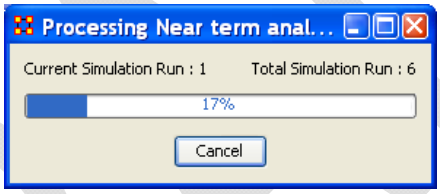


Figure 52: simulation processing procedure

4) Results from Near Term Analysis

The simulation results will be displayed in a new window 'Near term analysis result', Figure 17. Because we have two removal experiment cells, we have two performance lines over time. The lines represent the deviation from the baseline in terms of a performance measure, 'Knowledge diffusion'. Examining this performance change over time reveals how much the removal of the agent will impact to the performance of the organization. Also, we can draw a bar chart by clicking 'Draw bar chart' button (Draw bar chart). The bar chart, Figure 18, will display the performance comparison at the end time of the simulation.

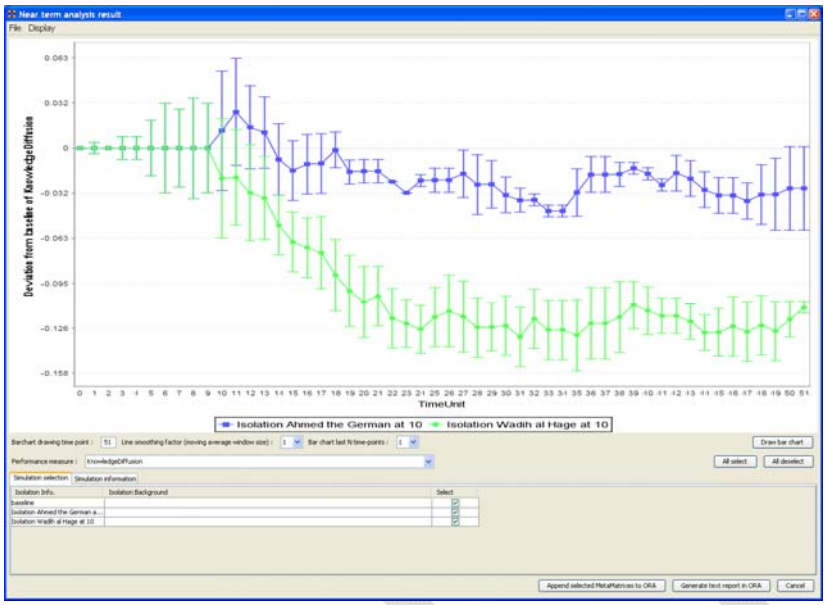


Figure 53: a line chart from Near Term Analysis displaying the performance over time

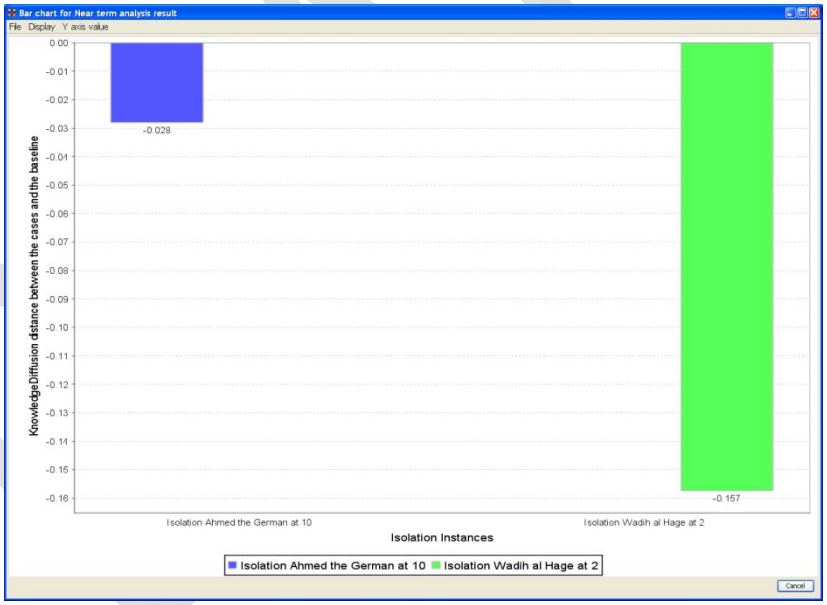


Figure 54: a bar chart from Near Term Analysis describing the deviation of performance at the end time

We can obtain just the displayed charts, Figure 20 and 21, by putting the chart in the clipboard of Windows. You can find a menu, 'Copy the chart in the clipboard', Figure 19, and it will take the displayed charts in the clip board, so you just paste in any Word Documents or Graphics editing tools.

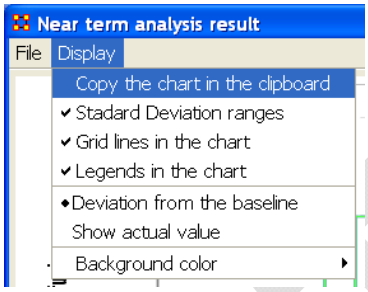


Figure 55: a menu for copying the images into the windows clipboard

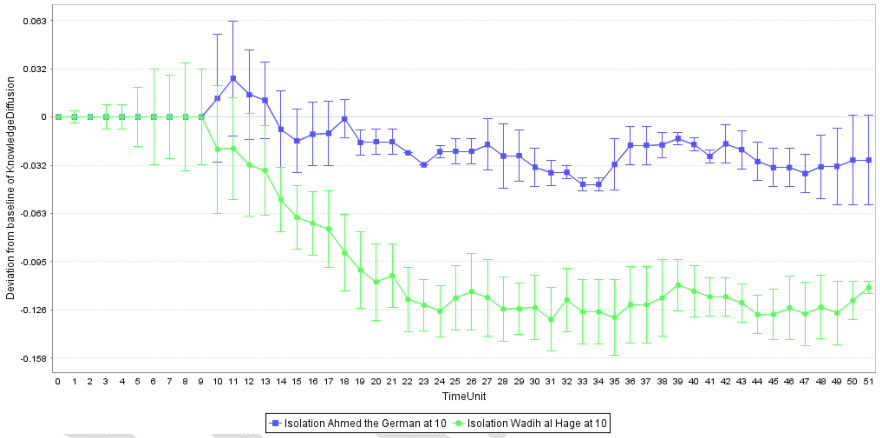


Figure 56: copied line char image

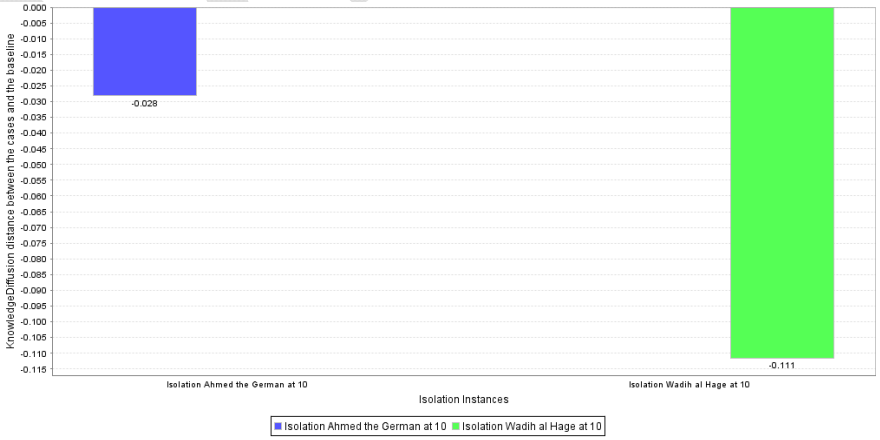


Figure 57: copied bar chart image

NTA produces three performance different estimations: knowledge diffusion, binary task accuracy and energy task accuracy. You can change the Y-axis performance value by selecting one of the three performance metrics from the drop down box at the middle of the result window, Figure 22.

Figure 58: performance metric change

The text output is available in HTML format. You can save the result HTML document by clicking 'Save the html report' menu under 'File' menu, Figure 23. After the click, you will be asked to provide a file name, the width and the height of images, Figure 24 and 25. If the HTML report is successfully saved, NTA will pop up a dialog saying that the document is well saved, Figure 26. The produced HTML report will look like Figure 27.

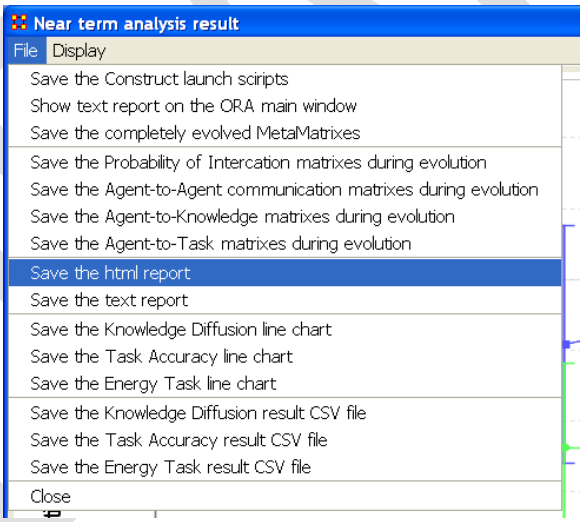


Figure 59: a menu for saving the HTML report

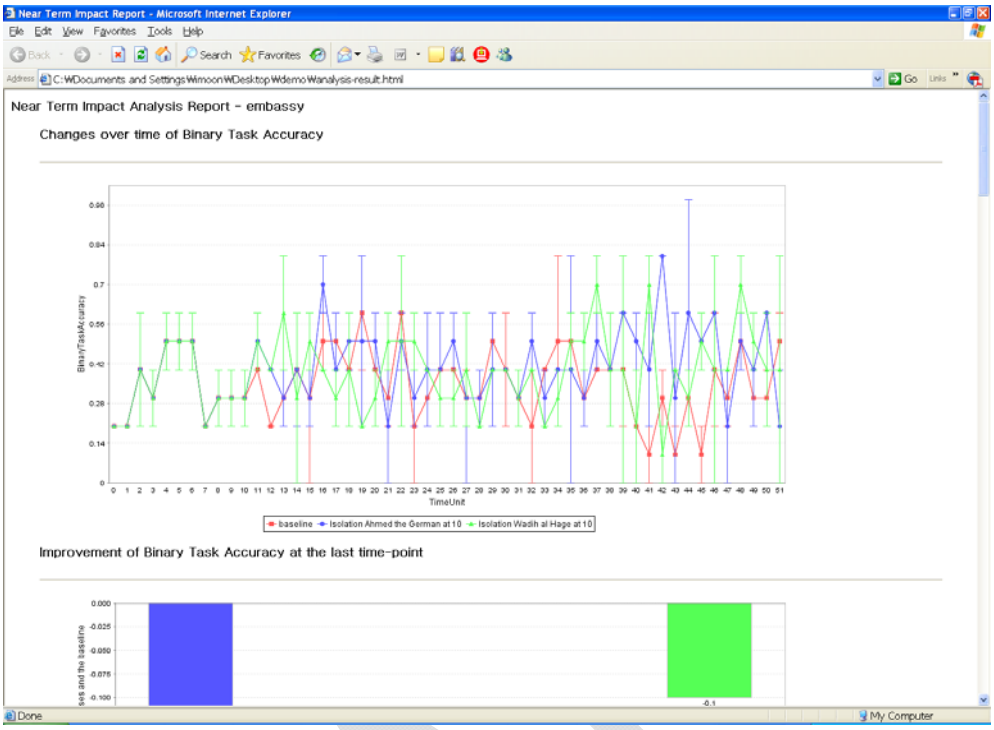


Figure 60 the saved HTML report

Finally, you can save the emerged organizational structures after simulations in the ORA main window. Under 'File' menu, there is a set of menus about saving different emergent structures. This demonstration will save the Agent-to-Agent communication network at simulation time step 40. First, you can find 'Save the Agent-to-Agent communication matrixes during evolution' menu under 'File' menu, Figure 28. If you click it, NTA shows a warning, Figure 29, that the saved networks will be from the simulation cases that you selected. You can select or deselect a simulation case by changing check boxes in the simulation case table at the bottom of the NTA window, Figure 30. If you click okay at the warning, NTA will ask you which time step you want to save, so we specify time step 40, Figure 31. If saving is successful, it will produce a popup dialog. After this, you will be able to see additional saved meta-matrixes in the ORA main window, as in Figure 33, so you can run additional ORA reports on the saved networks.

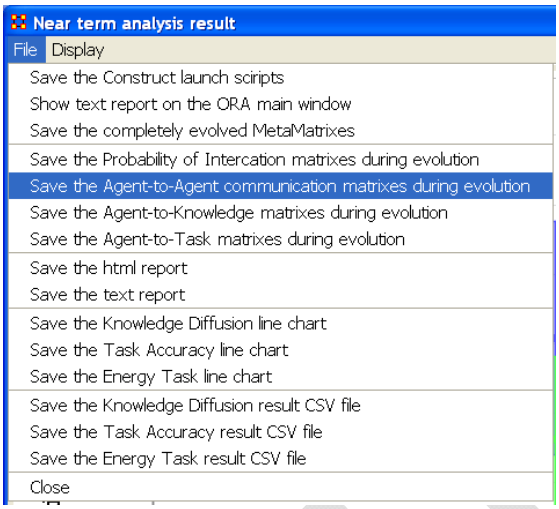


Figure 61 a menu for saving the evolved organizational structure

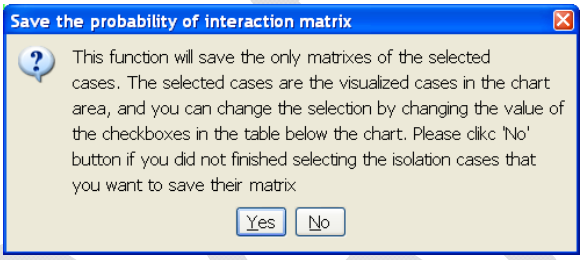


Figure 62 a warning about selecting cases to save

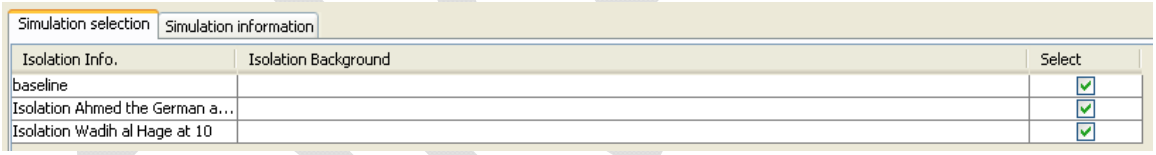


Figure 63 the simulation case list for selection

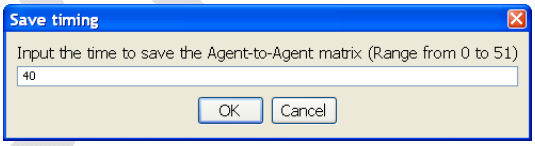


Figure 64 the simulation time-step to save

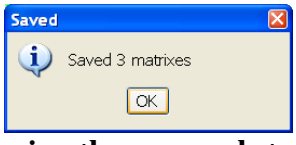


Figure 65: success message about saving the emerged structure

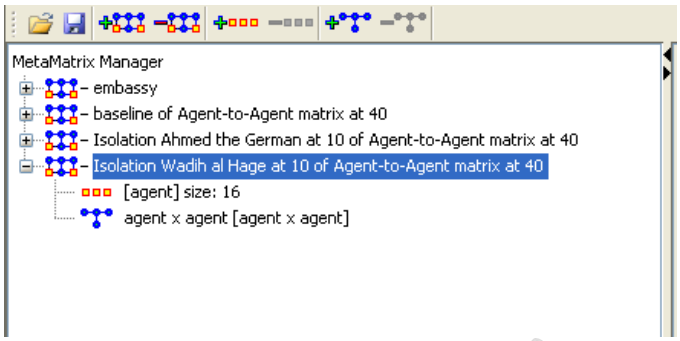


Figure 66 ORA main window showing the saved meta-matrixes

Moving Forward

ABC Company should now be able to gain valuable insight into how what they do impacts every other aspect of the corporation. They will know if "John" should be removed what the likely impact to the organization that will be the result. Any weaknesses inherent within the organization will likely be revealed. We have learned this through the demonstrated power of dynamic network analysis.

In the case of our C.I.A. analyst, we see that conducting immediate impact and near term analysis experiments that the isolation or removal of several agents is likely to impact the network in such a manner. It is there results that are key interest to the dynamic network analyst and by extension the C.I.A. analysts. How valuable would such information be? It could make or break an operational decision to carry out strategy. Perhaps it might totally alter tactics and this alludes to the one of the true power of dynamic network analysis: the power to accurately predict how a network will grow or perhaps even react to a certain situation.

Then we come back to the computer network administrator who studied the immediate impact of removing certain servers from the network. What was the result ot the over all efficiency of the network? These are all questions that the finger tips of the dynamic network analysts.

However, what we should be seeing by now, is that the ties that bind are constantly in flux and true link analysis that will prove more valuable to the network analyst is one in which a as many complex factors are included. We have learned that many networks, especially real ones, are multi-modal, that is they are not merely agents connected to agents, but real networks, involve different entity classes tied to other entity classes and that all of the entity classes taken as a whole, rarely stay the same. They evolve over time.

We can see that impacted such networks involves using tried and true scientific methods to calculate how certain networks will react and evolve over time given certain intervention or isolation strategies. It is this greater picture that proves the greatest information as to the true structure of complex real time networks.

In todays every sophisticated day and age of powerful computers and technology, our tools to analysis networks has become even more critical. Predicting how networks evolve could therein contain the keys to some of the world's most vexing problems, such as terrorism and even global warming.

Moving forward we have learned that networks inexorably move forward. Like the ticking of the second on the hand of an old clock, the ties that bind, continually move. Blink your eye and you just might miss the connection. Open them and you will see that dynamic network analysis is a tool that can help the network analyst in the ever increasing task of understanding the evolution and dynamics of real networks.

Now, we stand on the threshold of a new way of analyzing information wit the countless data that we are inundated with. We are using the power of powerful mathematics sand computer speed to make since our what that very computing speed has largely resulted in – an ever expanding and complex world that is only getting more complex and more expansive.

Will ABC Corporation learn how to thrive using the tools of Dynamic Network Anaysis? Will the management team better be able to undetsand the nature of the organztoianl that is called ABC Corporation? Will they know realize that "John" is a critical gatekeeper and that he his knowledge far exceeds his pay grade, not to mention his industry connectiosn?

Will the C.I.A. finally be able to infiltrate Al Qaueda and use the inherent weaknesses and strengths of the celluar network against its on mememberss? Will it spell the end for terrorism?

Will the computer server administrator finally streamline the servers in the manner that is most efficient for the users that need to draw on the vast computing power at their disposal?

In each and every case, the answer is that with Dynamic Network Analysis anything is possible when it comes to the ties that bind.

The field of Social Network Analysis itself is evolving. The questions managers and analysts are asking are maturing from the relative straightforward "what is the network", to the more complex and multifaceted, "what will the network become." Analysts no longer need to be just well-versed in network terminology, an assortment of measures, and evaluative interpretation; analysts now must add a deep understanding of dynamic theory and computer simulation to their repertoire. In order to be able to answer the "what will the network become" questions, the unit of analysis shifts from a single meta-network, to a series of meta-networks in the form of a time series. It would be somewhat unusual to be interested in the network forecast at a single point in time in the future, though for some reason, a single, forecasted meta-network may be of interest, regardless.

The analysis of evolving networks introduces the need for learning experimental design for computer simulation, social theory, and time series analysis. This chapter will introduce you to these three important dimensions and, with the help of ORA, walk you through running actual experiments involving social networks. We will take you though this chapter by presenting the particular approaches currently used at CASOS. There are certainly other approaches to forecasting evolution in networks, such as p*, that depend on the research question being addressed. We find that the approach we interoduce you to here will prepare you for not n=only the CASOS specific approach, but for understanding these other approaches, and maybe an approach you develop on your own!

The simulation tool used in this chapter is Construct (Carley 1995; Hirshman and Carley 2007; Carley, Martin et al. 2009; Hirshman and St. Charles 2009). Construct has three major tenets: the sociological principle of homophily (the notion that like seeks like), the social psychology principle of

transactive memory (the notion that perceptions may be more important than truth), and the principle that interactions may be subject to non-local causes (the notion that some interactions occur due to the presence or absence of distant third parties). Each of these principles, individually and in concert, can affect how networks evolve over time.

As a trivial example which illustrates these principles, consider a network of high school classmates. What would explain the pattern of friendship evolution measured over multiple measurements in a year? For instance, Mike and Jack might realize that they both have multiple common interests – both like tennis, Billy Joel, and Dungeons and Dragons. These common interests may lead Mike to form a friendship with Jack over the course of a year. Such an evolution would demonstrate homophily in action, as the common interest encourages the pair to hang out with each other after school. However, the fact that this friendship evolved, and was not present immediately, is a testament to transactive memory. At the start of the school year, Mike may not know about these shared interests because he might not know Jack, a transfer student. Over time, as Mike builds up a richer transactive memory representation of Jack's knowledge, Mike may begin to see that the two have much in common. This growth in transactive memory will feed into the evolving homophily process and likely lead to more frequent interaction. However, the budding friendship will be fundamentally embedded in a system where distant actions will have strong local effects. For instance, the fact that Jack's family moved into the school district may be due to his dad's new promotion – a factor that is independent of any attributes of either high school freshman. This promotion may have been due to the fact that his dad gave a wonderful presentation, which led the company to start a new office in the neighborhood … and so forth. What is important to acknowledge, though, is the fact that there may be a hundred kids in the city with attributes like Jack's, but the fact that Jack showed up in Mike's district and thus is available to make the tie is due to millions of complex and highly non-local actions. Thus, an evolving friendship between Mike and Jack must be appreciated as a rich and complex process.

Each of the principles underlying Construct has been examined by a variety of researchers in diverse fields. For instance, homophily has been widely recognized as an organizing principle for human societies for a substantial period of time (e.g., McPherson, Smith-Lovin et al. 2001). Homophily may occur along a variety of dimensions and in a variety of settings, though most work with homophily has focused on socio-demographic dimensions. For instance, substantial research suggests that gender homophily suggests that boys are more likely to associate with boys in elementary and middle school. Homophily along multiple dimensions may be possible, since a pack of high school boys interacting with each other will be homophilus according to age and education as well as to gender. However, homophily need not be limited to socio-demographic attributes. Instead, homophily can occur due to common interests, shared experiences, and other "deep level" attributes (Harrison, Price et al. 1998). Indeed, as groups of individuals evolve, they often move from interactions based on surface characteristics to interactions based on more fundamental, core similarities which may not easily be immediately observable. While it is important to acknowledge that certain situations a social network may evolve due to heterophily (the attraction between two very different types) such as might be expected among men and women at a bar, such special cases are more often the exception rather than the rule and can often be explained by a deeper similarity between initially different groups (in this case, perhaps, homophily along the "looking for a date" dimension). Due to the principle of homophily, then, one would expect a social network to evolve in such a way that individuals with common attributes, knowledge, beliefs, or other salient features would be more likely to interact in future time periods.

In our discussing homophily, we have already observed that there are multiple dimensions – some of which are not immediately obvious – along which homophily may occur. However, it is necessary to acknowledge that there are dimensions along which users do not realize they are homophilous: for instance, when individuals initially interact based on surface-level attributes, it is quite common for them not to perceive the deep-level attributes of their new partners others. This absence of deep-level knowledge harkens back to the idea of bounded rationality by which individuals take actions based on

their perceptions as opposed to an objective truth known by an omniscient observer (Simon 1957). More recently, this phenomena has been expanded and studied as the principle of transactive memory: the notion that individuals interact and form a shared memory unit in such a way that one individual may know that other individuals are knowledgable about some topic even without direct knowledge themselves (Wegner 1986; Carley 1991). This transactive memory often takes a substantial amount of time to build up, but can fundamentally shape the interactions between people. Specifically, transactive memory can be used to represent knowledge than an agent does not know. For instance, a husband may not be good with phone numbers but may remember that his wife knows them. Transactive memory can also be used to represent the fact that knowledge can go "out of date" – a one can remember that a friend was working for a large firm even though she was fired a year previously. These transactive memory perceptions are what can lead to fundamentally different types of interactions than would be expected if individuals were omniscient: if an omniscient salesman wanted to make a sale to the previously-mentioned large firm, he would probably not want contact the friend to facilitate the sale (and thus no link should be observed in a network). In the presence of imperfect information and potentially flawed transactive memory, however, such links can and will occur in an all-too-human network.

A final effect that has been explored primarily by researchers examining emergent and chaotic systems is that of the importance of small, non-local effects (e.g., Gladwell 2000). Such findings have suggested that a large event may be due to the confluence of a number of smaller events, none of which individually could have caused a specific outcome but which in aggregate can lead to substantial changes. Other researchers have begun to examine the statistical properties of large systems in order to understand how emergent behavior of small factors can combine to lead to a complex system (e.g, Epstein and Axtell 1999). Such research has suggested that some large, macroscopic behaviors may be due to the aggregations of many small changes over time, but it has also indicated that while the aggregate behavior of a system may be able to be predicted with some confidence the confident prediction of individual behavior may not be possible. Thus, any simulation which attempts to use a number of rules to forecast the behavior of a system must contend with the fact that predicting future behavior is, in a word, hard! While it may be possible to use homophily and transactive memory to guide how agents will interact in order to forecast network evolution, it must always be recognized that seemingly "random" effects may lead a network to evolve in a certain direction and can have profound effects upon overall outcomes.

While the above principles have been separately explored by researchers for a number of years, Construct is among the first generation of tools which has sought to unite the three together to study network evolution. This process of unioning these ideas, however, has been understandably complex. Therefore, it has been necessary to validate the model by comparing simulation results with case studies that have occurred in the real world. While the validation of Construct is an ongoing process, as the tool is constantly being improved, several validation case studies are worth mentioning. For instance, the original interaction model was validated using interaction data from Kapferer's tailor shop in Zambia (Kapferer 1972; Carley 1991). Construct has also been used to understand how information diffusion occurs in the presence of "smart agents", such as databases (Carley 1999) as well as on educational interventions such as web pages (Carley, Martin et al. 2009). Other work with Construct has examined the tiering effects that occur in social networks, such as occur when individuals are close to a small group of people but relatively distant to a larger group, and have found correspondence between simulated societies and real ones (Zhou, Sornette et al. 2005; Hirshman and St. Charles 2009).

It is important to note that other authors have identified alternative techniques for understanding evolving networks, including techniques that emphasize the transitivity and reciprocity of network relationships, or techniques that take into account specific network properties of the networks in question. For instance, models such as Sienna (Steglich, Snijders et al. 2006) choose to weight a variety of dimensions in determining network evolution. Construct is specifically an agent-based model, which allows individual actors to store knowledge, interact, learn, and modify their behavior in order to evolve the network. The Construct model presented in this chapter represents one of multiple ways in which

researchers have begun to explore the evolution of networks. The three dimensions presented above have been proven in a number of settings and have been employed to some degree in many network forecasting models, sometimes as part of agent-based models and sometimes as parts of a different representation.

The graphical user interface for Construct , ORA's Near Term Analysis tool (Moon and Carley 2007), will allow a beginning user to harness some of the simulation power available to Construct and thus can serve as a gentle introduction to network evolution. The Near Term Analysis tool employs the Construct simulation engine in order to forecast the evolution of a network according to known social and psychological principles. The Near Term Analysis tool is a graphical user interface for interacting with Construct and can greatly assist the process of experiment setup, analysis, and debugging. The Near Term Analysis has been a part of the ORA package since <WHEN> in <YEAR>.

The data (1 sentence)

Use the book's standard dataset

Topics (12-16 pages)

DNA Measures (6-8 pages)

Relative similarity

Relative expertise

Agents are decision-makers with varying information processing, socio-demographic, and access constraints and as such may or may not be human (Carley, 2002). Within Construct, agents go about their business interacting, communicating and learning each time period, as described in Figure 1. As agents learn or acquire information, they may change their preferred interaction partners and modify what they are likely to communicate. These factors, in turn, influence what types of decisions are made by each agent. A variety of factors influence who agents select as interaction partners, what they communicate with that partner, how much and how they communicate, whether they learn anything from that partner, and the accuracy and sustainability of that learning. Such factors include the agent's socio-demographic characteristics, information processing characteristics, proximity, and current position in the social and knowledge networks. The agent model has been described in depth in other venues (e.g., Carley, 1990 & 1992; Hirshman, Carley & Kowalchuk, 2007a & 2007b); thus, we concentrate here on both a high level description and details of those components used for the simulations reported.

Within Construct, agents both influence and are influenced by others. Agents who have influence over others can use that influence to escalate or de-escalate activity at a societal level by communicating information and/or beliefs. Social influence – as derives from shared attributes such as socio-demographic factors, shared knowledge, beliefs, and proximity – co-evolves with the spread of knowledge and beliefs (Carley, 1991). Consequently, in more heterogeneous populations where the lines of differentiation line up the chance of self- reinforcing beliefs at the group level is greater (Blau, 1977). Factors that are not influenced by the diffusion of information and beliefs include the agent's socio-demographic role (e.g., age, race, gender, level of education), the agent's basic cognitive limitations and information processing capabilities (e.g., likelihood of forgetting, risk taking, amount of information and beliefs that can be communicated or processed, and whether the agent has transactive memory), the size of their sphere of influence (at least in the short term), and factors that have resulted from socio-cognitive interactions (e.g., literacy, access to newspapers, radio and the internet).

Within Construct, agents develop likelihoods of interacting with others based on relative similarity (RS) and relative expertise (RE) (Carley, Lee & Krackhardt, 2001; Hirshman, Carley & Kowalchuck, 2007a). Relative similarity is a homophilly based mechanism (McPherson and Smith-Lovin 1987; Carley 1991) and derives from the idea that individuals are more likely to interact if they have more in common. Homophilly based interaction is a multi-causal phenomenon due to ease of communication, shared understandings, and comfort. The relative similarity of i and j, from i's perspective, is characterized as

〖RS〗_ij=(∑_(k<K)▒〖〖(AK〗_ik*〖AK〗_jk)〗)/(∑_(j<I)▒∑_(k<K)▒〖〖(AK〗_ik*〖AK〗_jk)〗)

where individual i's relative similarity to j, is determined in terms of socio-demographics, knowledge, and belief items K in the agent-to-knowledge matrix AK.

Of important note: an individual is most relatively similar to itself, and each period will have a reasonably high probability of choosing to "interact with itself" and to avoid communicating with others. Just because an agent has the highest relative similarity with itself, however, does not mean that an agent will always interact with itself; indeed, due to the large number of other agents in the simulation, such avoidance of communication is relatively rare.

Relative expertise is a search based mechanism and derives from the idea that individuals are more likely to interact if one has information that the other wants. The relative expertise of j as judged by i is characterized as

if 〖AK〗_ik=0,then X_jk=〖AK〗_jk else X_jk=0

〖RE〗_ij=(∑_(k<K)▒X_jk)/(∑_(j<I)▒∑_(k<K)▒X_jk)

where individual i's relative similarity to j, is determined in terms of socio-demographics, knowledge, and belief items K in the agent-to-knowledge matrix AK (Schreiber 2006).

Agents are more likely to initiate interaction with another if they think the other has information they need and/or they are similar to them. However, there is a curvilinear relation between this familiarity and expertise; to wit, as agents initially increase in similarity (homophily) they are more likely to realize the other has expertise they need but as they increase still further in similarity they realize that the other is so similar there is no specialized expertise.

The researcher needs to specify the strength of each of these factors for agent-agent interaction. Herein, we set all human agents to use both logics and to at any time create a combined probability of interaction that is based on 60% similarity and 40% expertise. In both cases, individuals are giving and receiving information and the overall tendency to give versus receive is about 60/40 as identified by Valente, Poppe and Merritt (1996).

When setting up a virtual experiment in Construct the researcher needs to specify multiple parameters for each agent. This is often facilitated by the used of agent classes to parameterize multiple agents simultaneously. We next discuss: the number of agents in each of the classes of this experiment (section Error! Reference source not found.), the distribution of socio-demographic parameters for the agents of that class (Error! Reference source not found.), the distribution of cognitive factors for each class (Error! Reference source not found.), the sphere of influence for that class (Error! Reference source not found.), and the access constraints for that class (Error! Reference source not found.). While the full Construct model has a number of features that can be varied, such aspects were held constant for this simulation (Hirshman, Carley & Kowalchuck, 2007a).

Knowledge similarity and knowledge expertise, diagramed in Error! Reference source not found., are fundamental to Construct's operation (Hirshman and Carley). Both similarity and expertise calculations rely on an agent's transactive memory and are fundamentally cognitive in nature. For instance, if an agent knows a fact, and the ego knows that the potential alter agent also knows it, then the fact contributes to a perceived similarity between ego and alter. In actuality, the alter may or may not know the stylized fact, or could possibly have known the fact and forgotten it. However, the ego agent perceives the similarity, and the perception leads to increased likelihood of interaction between the two agents. If the ego agent does not know the fact but perceives that the alter agent knows the fact, then the ego considers the alter to be a "relative expert" in that area, which increases the ego's probability to interact slightly (Hirshman and Carley). Note that if the ego knows that the alter does not know the fact, or if the ego has no transactive memory of the alter's knowledge, then the probability of interaction will be unaffected.

The algorithm outlined in Error! Reference source not found. is run for all facts for one ego/alter pair of agents, resulting in an overall score for that potential interaction partner. The agent then chooses to interact with an agent based on this probability score, an action that is fundamentally social in nature. It should be noted, however, that increasing similarity or expertise may not lead to an increase in interaction between two agents. If an ego agent learns a fact from an alter, the two agents will have increased similarity on an absolute scale; however, the same fact may cause the ego agent to become more similar to many other agents, thus leading to a decrease in relative similarity. Thus Construct, like human behavior, is highly non-linear and agent-agent interactions are as dependent on the similarity between agents as they are the differences between agents and other potential interaction partners.

Construct Cycle (6-8 pages)

As mentioned previously, Construct is an agent-based model. This means that individual actors in Construct will individually evaluate their options and make decisions. For the purposes of this chapter, the decisions will be of the kind "who should I interact with?" although agents in Construct are able to perform much more complicated decisions based on their knowledge. As agents interact in Construct, the network on which the agents operate will gradually evolve. As agents choose new interaction partners, interact more frequently with some old partners, and drop others, the network itself will evolve. When this process is evaluated over all the agents in the network, profound shifts in network behavior can be observed.

In order to understand how Construct goes from the theory (described at the beginning of this chapter) and the calculations of relative similarity and expertise (described in the previous section) to changes in the overall structure of the network, it is necessary to understand how agents in Construct interact. While we present an overview of the five-step Construct interaction cycle in this section, we abridge it slightly so as to only focus on the outcomes which directly influence network evolution. Additional details about Construct, as well as information about its full capabilities, can be found in several technical reports about the tool (Hirshman and Carley 2007; Hirshman and Carley 2007; Hirshman and Carley 2008) or the project website, http://www.casos.cs.cmu.edu/projects/construct/.

Briefly, the Construct decision cycle is as follows. Each interaction cycle consists of the five sub-processes described in Figure 1, as read counter-clockwise starting from the top:

Figure 1: The Construct Interaction Cycle

1. Agents evaluate their potential interaction partner using the relative similarity and relative expertise factors described earlier.
2. Each agent select an available agent with whom to interact using the scores derived from the potential partner evaluations.
3. Every agent communicates with its selected interaction partner; both the interaction initiator and the receiver send a message to the other.
4. Both agents have the potential to learn new information from this communication.
5. Agents update their knowledge and perform any actions specified in the simulation. This ends the time period, and a new time period starts.

It is worthwhile to delve into this cycle in slightly more detail in order to understand where homophily, transactive memory, and non-local effects can affect the simulation.

At the beginning of each time period, agents first employ their transactive memory to rank their possible interaction partners according to homophily-based similarity. Agents do this by creating a "probability of interaction" with each potential partner, a relative score which reflects the chance that any particular agent will be chosen (Hirshman and Carley 2007). This probability of interaction is affected by multiple factors: knowledge similarity and expertise, as described earlier, which are calculated using homophily and transactive memory; socio-demographic similarity (Harrison, Price et al. 1998; McPherson, Smith-Lovin et al. 2001; Hirshman, Martin et al. 2008), physical proximity, which is similarity due to similar positions in the physical environment (i.e., Barnlund and Harland 1963; Butts and Carley 2000), and social proximity, a catch-all for other types of similarity not explicitly modeled. While knowledge similarity and knowledge expertise must be present in all Construct simulations, the other forms of similarity are optional and can be enabled by the experiment designer. In some cases, such forms of similarity may prove to be extremely important; in others, it may be simpler to omit the factor and run the simulation based purely using knowledge as the driving factors. Regardless of which factors are used, the end result is effectively a score, for instance that an agent A will be twice as high a probability of interaction with agent B as with agent C, but three times as high a probability with agent D. This example is illustrated in Figure 2, and will be used as an example for the remainder of this explanation.

Figure 2: Interactions in Construct

<DIAGRAM FORTHCOMING>

Once agents have computed these rankings, each agent chooses its interaction partner or partners. Since there are usually a large number of agents in the simulation, it is not fair (or realistic) for one agent to choose first every time period. Thus, the order of partner selection is randomized. This means that while the interaction order may be A, B, C, D one period, it may by B, D, A, C the next and a still different order in a subsequent period. This interaction order can have subtle effects on partner selection. Consider the example from Figure 2, where agent A has the highest probability of interaction with agent D. If agent D is busy (perhaps already has been selected by another agent as an interaction partner), then agent A has to choose between agent B and C, and will likely choose the former since it has the higher probability. However, if A has the opportunity to choose first, it will likely want to select D. Note that the availability of the highest-ranked partner will not guarantee its selection: agents will use the probabilities of interaction in order to weight partner selection. Due to the randomness introduced in the process of interaction order and partner selection, Construct inherently introduces non-local effects into the simulation process: by making another agent select D as a partner before agent A has a chance to go, we can remove agent D from the list of A's possible interaction partners and thus affect the evolution of the network in that time period and later ones.

The partner selection process continues until all agents have selected available partners or choose to "interact with themselves" if no partners are available. Once all agents have partners, both the interaction initiator – the one who chose the interaction partner – and the interaction receiver – the one who was chosen – prepare a message to send to the other partner. Note that two messages are prepared, as communication in Construct is a two-way street. Both messages are drawn from the sender agent's current knowledge. These messages consist of a sequence of facts, beliefs, or transactive memory items regarding third parties. Thus, depending on simulation parameters, Construct agents have the potential to send informative statements ("a knight on the rim is dim"), beliefs on issues ("I believe that you should move your queen's pawn first"), and transactive memory about others ("agent B from Figure 2 is the local chess expert"). While the exact composition of the message is subject to cognitive limitations as well as item weighting (see, for instance, Hirshman and Carley 2007), the message sent will always be a subset of what the agent actually knows. The type of message transmitted represents another area in which random and potentially non-local effects can be introduced to the system, since agents can re-transmit facts learned from third parties. Thus, the very nature of the message sent by the agent can fundamentally affect interactions of distant third parties many time periods after the initial message has been sent.

Once both agents have prepared messages, the messages are sent from one agent to the other. At this point, agents have the opportunity to learn new information about their communication partner. For instance, if the other agent sent a fact, the receiver will have the opportunity to learn the fact – but then will also be able to create the transactive memory which states "agent C from Figure 2 sent me this fact, therefore he knows the fact" as well. Note that some agent cognitive limitations (e.g, Hirshman and Carley 2008) may affect the message elements learned by the recipients and can have profound effects on what agents actually perceive about their environment. Regardless of what the agent actually learns, the internal state of the agent will change slightly. This new knowledge may lead the agent to change its probability of interaction in future periods, as it realizes that it is relatively more similar to one agent than another. The knowledge may even make the agent relatively less similar to its old interaction partner, as the facts or beliefs transmitted may decrease the probability of interaction between them (or greatly increase the probability of interaction between the agent and multiple other third parties, leading to a decrease in the original agent-partner relative probability of interaction). In this way, the learning is fundamentally intertwined with the evolution of the system as the changing knowledge drives the change in the observed network among agents.

At this point in the Construct cycle, agent interaction is complete. What remains is to gather statistics and update the simulation, as well as to allow agents to perform decisions (e.g., Hirshman, Carley et al. 2009). For the purposes of studying the evolution of an agent-to-agent social network, it is necessary to figure out who interacted with whom in order to store that information for subsequent analysis. However,

it may also be useful to store who knows what information, an agent-to-knowledge network, in order to better understand why two agents were drawn to each other or to understand the rate at which knowledge similarity is increasing over time. The Near Term Analysis tool provides a snapshot of many of these networks in order to facilitate analysis and to help the beginning user understand how networks are changing over time. Once these statistics have been gathered, the next simulation period starts and the network evolution continues.

Attributes

According to relationships/links

Walkthrough Near-term impact report (6-8 pages)

Show how to set up experiment

Set up experiment parameters

Indicate the number of replications

Indicate the number of time periods

We can examine the impact of removing a particular entity from the network by using Near Term Analysis (NTA). NTA removes a given entity and estimates the changes in the remaining network.

This provides a what if?... analysis of the network. We can simulate strategic interventions or eliminations of certain agents, and examine how the network should react and change. For example, we can look at how the Tanzania Embassy bombing network changes if Ahmed Ghailani and/or Wadih El-Hage are eliminated.

To run a Near Term Analysis:

1. Start the Near Term Analysis window

· Return to the ORA interface.

· Make sure the meta matrix is highlighted in Panel 1. If it is not highlighted, single-click on it.

· Go to Analysis in the menu bar and select Near-Term Analysis. A window titled Near Term Analysis will pop up.

NTA has two modes, Novice and Advanced, displayed as tabs in the Near Term Analysis pop-up window. Novice mode provides an automatic analysis set-up, but does not allow users to set up their own hypotheses.

· Click on the Advanced mode tab to switch to Advanced mode.

Below is a screen capture showing the Advanced mode tab in the Near Term Analysis pop-up window:

In the field titled Settings, find the number boxes on the far right. Change the number of replications (the top number box) to 2, and the number of simulated time-points (the bottom number box) to 52. The timeline in the Simulated time line field will expand to include 52 time-steps.

2. Open the simulation scenario wizard

· In the middle of the pop-up window, find and click the Add new simulation instances button. A small window titled Near Term Analysis - Simulation scenario create wizard will pop up.

Below is a screen capture showing this wizard window:

In this wizard window, make sure the first option in the Options field is selected. Click the Next button. The wizard window will display options for creating a custom scenario.

59

Below is a screen capture showing this custom scenario field in the wizard window:

59

Below is a screen capture showing this custom scenario field in the wizard window:

3. Remove Wadih El-Hage

- Click the box for Wadih El-Hage in the table of agents in the wizard window.
- Change the Timing box to 10 in the same row.
- To add this event set-up to the Near Term Analysis, click the Add events button
- located next to the Search field (above the list of agents).

The timeline at the top of the wizard window will mark the isolation of Wadih El-Hage at time 10.

60

Below is a screen capture of this timeline:

Click the Add a simulation button at the bottom of the wizard window. The wizard window will close automatically. In the NTA window, the new simulation case will appear in the Cases to simulate field.

4. Remove Ahmed Ghailani

· Follow Step 3, but replace Wadih El-Hage with Ahmed Ghailani.

Your two simulation cases appear in the Near Term Analysis window in the Cases to simulate field, along with the Baseline case. The Baseline case is simply the network with no entities removed.

You can view your different simulation cases by clicking on them in the Cases to simulate field, and the simulation time-line will change accordingly.

5. Run the Near Term Analysis

· Click the Execution button at the bottom of the Near Term Analysis window.

· A small window will pop up to warn you that the execution may take a long time. Click OK. Another small window will pop up to show you how the simulations are progressing in the Near Term Analysis.

6. View the results of the Near Term Analysis

· When the analysis is complete, a new window titled Near Term Analysis Results will pop up.

61

Below is a screen capture of the results pop-up window:

These two performance lines correspond to deviations from the baseline of knowledge diffusion over time. Examining this performance change over time reveals how much an agent's elimination impacts the performance of the organization.

· To draw a bar chart, click the Draw bar chart button in the results window. The resulting bar chart displays a performance comparison for each time-point.

62

Below is a screen capture of the bar chart:

The results of the Near Term Analysis are also available in HTML format.

7. Save the results of the Near Term Analysis in HTML format

· Go to File in the menu bar and select Save the html report. A file chooser window will pop up. Navigate to the location you want and type a filename for your HTML report.

· A window titled Image Size Input will pop up. Type in the width of the image and click OK.

· Another window titled Project Saved will pop up displaying the location of your saved analysis results. Click OK.

8. Save the organizational structures resulting from your simulations in ORA

· Return to the ORA interface.

· To save the Agent-to-Agent communication network at time-step 40 in the simulation, go to File and select Save the Agent-to-Agent communication matrices during evolution.

· A small window titled Save the probability of interaction matrix will pop up. Click the Yes button.

· Another small window titled Save timing will pop up. Enter 40 and click OK.

· A final window titled Saved will pop up. Click OK.

63

You can view saved simulations in Panel 1 of the ORA interface.

Set up the baseline and scenarios

Baseline is automatically included in the initial set up

Set up other scenarios

Indicate which node(s) and when the isolation(s) should occur

Show how to read output charts

The output charts indicate the performance and the knowledge diffusion over time for each of the scenarios and the baseline

Problem Set

What 5 factors are considered when Construct selects an interaction partner for an actor?

What is the minimum number of alters a specific agent can interact with in a given time period? What is the maximum?

What are the four types of data-t items that an actor can communicate to another actor?

Does an actor maintain transactive memory about himself?

Why does an analyst run more than one replication for an experiment?

What are the four node classes that represent an organization in construct?

In construct how is organization performance determined?

What does the measure knowledge diffusion indicate?

Can the construct script file that the near-term impact report creates, be used in the construct stand-alone software?

What is relative similarity?

In the dataset which actors/dyads have the greatest level of relative similarity (NEED TO GET ORA TO COMPUTE THIS???; maybe use the existing redundancy measures???)

What is relative expertise?

In the dataset which actors/dyads have the greatest level of relative expertise (NEED TO GET ORA TO COMPUTE THIS??? maybe use the existing exclusivity measures???)

What is the purpose of running a baseline scenario in the near-term impact report?

#1: isolate an actor

Run the near-term impact report for a 100 period (10 iterations) and isolate Ahmad the German at time period 20. How does the isolation impact the organizational knowledge diffusion before the intervention, then after the intervention? How does the isolation impact the organization performance, before and after?

#2: Isolate a knowledge node

Run the near-term impact report for a 100 period (10 iterations) and isolate marketing knowledge at time period 20. How does the isolation impact the organizational knowledge diffusion before the intervention, then after the intervention? How does the isolation impact the organization performance, before and after?

#3: Isolate two entity types

Run the near-term impact report for a 100 period (10 iterations) and isolate both Ahmad the German and marketing knowledge at time period 20. How do the isolations impact the organizational knowledge diffusion before the intervention, then after the intervention? How does the isolation impact the organization performance, before and after?

#4: Exploring options

Which if the following interventions are expected to have a greater reduction in organization performance at time period 85?

A) Isolate ahmed the German at time 20

B) Isolate ahmed the German at time period 40

C) isolate Osama bin laden at time period 20

D) isolate Osama bin laden at time period 40

E) Do nothing

CHAPTER 9: Detecting Change

ABC Corporation is now faced with a whole new task. They want to compare their organizational structure as a whole with exactly the same data they obtained several years back. So, how would ABC Corporation go about doing this? What methods would they use and what criteria would be the most important for ABC Corporation?

Would the organizational analyst have any particular idea given vast amounts of data about what change is really evident within the organization across a certain span of time? It would seem like a daunting task but the dynamic network analysts would step in an offer a solution to this problem.

What about the CIA and their task for combating terrorism such as an Al Qaeda cell. What if they now have several years' worth of complied data to examine in terms of what they are fighting now? What should their method be? Would there be a meaningful way to look at the mountains of information to detect sublet yet critical changes within a network's infrastructure? Perhaps the money is coming from a different source? Perhaps a new leader has emerged who is better at motivating certain others than the previous leader. Maybe the cell is more active. Perhaps it evolved into a denser organztion because of the pressures of the C.I.A over the several years. In any case, detecting change will be critical to establishing what methods the C.I.A. is employing that are actually working against Al Qaeda and which one's are failing. So how does the C.I.A. analyst go about such a comparison? Where would he begin and what approach would he take?

Let us revisit the computer administrator? Let us say that he is now interested in how server network usage has changed over the years. He wants to know a little bit more about user's of the network and how they go about their work. What are they doing differently now than they were doing previously? What else is in store for them? How will these changes affect the status of the network? There are many considerations and mountains of data to pour though. Is there a method that the dynamic network analyst could employ to help detect network change?

Change in Networks

Change in networks can be hard to detect when we are considering multi-modal network data. Primarily because there are innumerable ties and entities to deal with and such change often times does not become apparent until the network has changed into something dramatically different. For instance, in the case of an Al Qaeda cell perhaps the change inside the network is not apparent until what once was a benign network suddenly become very agitated and active. Then it becomes dangerous.

Measuring Change

So exactly, how does the network scientist measure the differences between two networks? Well, the short answer is that they dynamic network analyst is interested in learning about the "distance" between the network. To do so, is to consider the network as a mathematical string. Moreover, the distance between two strings can be calculated on a number of metrics. The most common means for doing so is to calculate the "Hamming" distance between two networks. The other methods are to use binarized data and Euclidean geometry to ascertain differences. Each method is done differently and warrants its own respective considered by the analysts. You can also used non-binarized data. Binarized data is data that has been converted into "Os" and "1s". For instance, we might say that "John" has a link with "Mary" and therefore we give this linkage a "1". If they did not have a connection, we would give it a "0". This is how data is binarized.

Hamming Distance Explained

The Hamming distance is named after Richard Hamming, who introduced it in his fundamental paper about error-detecting and error-correcting codes (1950). It is used in telecommunication to count the number of flipped bits in a fixed-length binary word as an estimate of error, and therefore is sometimes called the signal distance. Hamming weight analysis of bits is used in several disciplines including information theory, coding theory, and cryptography. However, for comparing strings of different lengths, or strings where not just substitutions but also insertions or deletions have to be expected, a more sophisticated metric like the Levenshtein distance is more appropriate.

In information theory, the Hamming distance between two strings of equal length is the number of positions for which the corresponding symbols are different. Put another way, it measures the minimum number of substitutions required to change one into the other, or the number of errors that transformed one string into the other.

For a fixed length *n*, the Hamming distance is a metric on the vector space of the words of that length, as it obviously fulfills the conditions of non-negativity, identity of indiscernibles and symmetry, and it can be shown easily by complete induction that it satisfies the triangle inequality as well. The Hamming distance between two words *a* and *b* can also be seen as the Hamming weight of *a*−*b* for an appropriate choice of the − operator.

For **binary strings** *a* and *b* the Hamming distance is equivalent to the number of ones in *a* xor *b*. The metric space of length-*n* binary strings, with the Hamming distance, is known as the *Hamming cube*; it is equivalent as a metric space to the set of distances between vertices in a hypercube graph. One can also view a binary string of length *n* as a vector in $\mathbb{R}^n$ by treating each symbol in the string as a real coordinate; with this embedding, the strings form the vertices of an *n*-dimensional hypercube, and the Hamming distance of the strings is equivalent to the Manhattan distance between the vertices.

The number of bits which differ between two binary strings: ∑ | Ai - Bi | alternatively, it can be calculated as Union (Ai & Bi) – Ai. Either formula works for weighted or binary data.

Comparing two binary matrices – number of edge flips to make B = A

Typically convert hamming to difference as a percent

Difference = 100*(Max_possible_distance – Hamming)/ Max_possible_distance

Max_possible_distance = N*N-1 (assuming 0 diagonals)

However, there are some important limitations to treating networks as statistical strings:

- There are row column dependency's
- In other words – each entry is a dyad and dyads are not independent
- The basic assumptions of standard statistics are violated
- Estimation procedures designed for independent observations will calculate incorrect standard errors

Moreover, depending on the statistical method employed to compare the networks, we have to think about other factors as well. These might be called approaches to the statistical models above:

Fixed Effects
- Would require dummy for each row and column
- May be inefficient or parameters may not be estimable

Random Effects (Generalized Least Squares)
- Requires modeling and estimating covariance matrix
- If model is wrong, estimates may be inefficient and standard errors may be incorrect

Empirical Standard Errors
- Use estimation procedure based on independence (e.g. OLS), but adjust standard errors
- In QAP, standard errors are estimated by using permutations of the data set

Therefore, what can what can the network analyst do to address some of these limitations?

- QAP – quadratic assignment procedure
  - Pearson correlation
- MRQAP – multiple regression quadratic assignment procedure
  - Regression on networks

- Qaptest tests an arbitrary graph-level statistic against a QAP null hypothesis, via Monte Carlo simulation of likelihood quantiles.
- Preserve row-column dependencies
- Non-parametric techniques for assessing similarity between networks
- A resampling-based method, similar to the bootstrap, for calculating the correct standard errors
- The null hypothesis of the QAP test is that the observed graph-level statistic on graphs *G_1*,*G_2*,... was drawn from the distribution of said statistic evaluated (uniformly) on the set of all relabelings of *G_1*,*G_2*,....
- This test is performed by:
  - repeatedly (randomly) relabeling the input graphs
  - recalculating the test statistic
  - evaluating the fraction of draws greater than or equal to (and less than or equal to) the observed value
- The accumulated fraction approximates the integral of the distribution of the test statistic over the set of unlabeled input graphs.
- See: http://www.maths.lth.se/help/R/.R/library/sna/html/qaptest.html
- Permutes the dependent variable only
- Permutes rows and columns the same
  - This is the step the preserves row-column dependencies

- Resulting matrix:
  - Corresponds to the null hypothesis
  - Preserves any row and column dependence of both dependent and independent variables

We can apply algorithms as well. Why would the network analyst be interested in doing just that? Let us look at the following reasons:

- Permute the dependent variable and merge back with the independent variable
- Run the estimation with the new merged data set, and save the result
- Repeat the permutation and estimation to generate an empirical sampling distribution

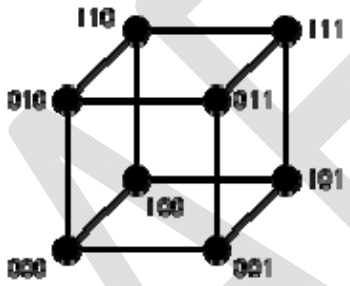


Figure 67: Hamming Distance Cube

Change versus random permutation

So how do we begin to interpret the data after we apply a certain statistical approach to comparing the networks and detecting change? Keep in mind that we don't simply have to be limited to two separate periods; rather we can compare networks of relatively equal consideration to each other in pursuit of network change as well. This in a sense is network difference.

- In how many of the saved runs was the observed >= new?
- That's the significance
- Typical – run 100 cases
- Note – the larger the network – the larger the number of permutations expected for same "coverage"

Final word on detecting change

Back to ABC Corporation we now have the dynamic network analysts in the fold. He or she will compare the organizational data form over several times by looking at the organizational data as statistical strings. We have learned this approach will carry both rewards and some drawbacks. The data doesn't work perfectly as a string of statistical relationships, but it can tell us quite a bit about detecting change in networks.

Our CIA analyst is particularly interested in detecting change. After all, he needs a method to compare differing network data capture over two different time periods so he or she can determine exactly how the networks differ and how they have changed.

The computer network administrator will have a keen interest in how his network of servers has changed over time. He needs an approach to measure that change and detect it. Converting the network over to statistical strings is one such manner to do this. There are limitations but rewards as well.

In the next chapter will consider how networks evolve over time. We will discuss methods for simulation the evolution sand what can be learned from such approaches. We will learn the effects of running a near term impact analysis, which considers network change after several key events have taken place to impact the network. We will see who the network analyst will evaluate such network evolution and what it can tell him about it.

CHAPTER 10: Prediction

The learning objectives of this chapter are:

(a) Be able to take networked relational data from a specific time, simulate an evolution of that network data based on realistic dynamics grounded in social theory, and forecast what that network might be in the future,

(b) Be able to conduct controlled experiments that can compare various forecasts developed under mixed conditions of social interventions in that network, and

(c) Be able to use ORA as a tool to construct these forecasts and conduct these experiments.

Literature background (1-2 pages)

The field of Social Network Analysis itself is evolving. The questions managers and analysts are asking are maturing from the relative straightforward "what is the network", to the more complex and multifaceted, "what will the network become." Analysts no longer need to be just well-versed in network terminology, an assortment of measures, and evaluative interpretation; analysts now must add a deep understanding of dynamic theory and computer simulation to their repertoire. In order to be able to answer the "what will the network become" questions, the unit of analysis shifts from a single meta-network, to a series of meta-networks in the form of a time series. It would be somewhat unusual to be interested in the network forecast at a single point in time in the future, though for some reason, a single, forecasted meta-network may be of interest, regardless.

The analysis of evolving networks introduces the need for learning experimental design for computer simulation, social theory, and time series analysis. This chapter will introduce you to these three important dimensions and, with the help of ORA, walk you through running actual experiments involving social networks. We will take you though this chapter by presenting the particular approaches currently used at CASOS. There are certainly other approaches to forecasting evolution in networks, such as p*, that depend on the research question being addressed. We find that the approach we interoduce you to here will prepare you for not n=only the CASOS specific approach, but for understanding these other approaches, and maybe an approach you develop on your own!

The simulation tool used in this chapter is Construct (Carley 1995; Hirshman and Carley 2007; Carley, Martin et al. 2009; Hirshman and St. Charles 2009). Construct has three major tenets: the sociological principle of homophily (the notion that like seeks like), the social psychology principle of transactive memory (the notion that perceptions may be more important than truth), and the principle that interactions may be subject to non-local causes (the notion that some interactions occur due to the presence or absence of distant third parties). Each of these principles, individually and in concert, can affect how networks evolve over time.

As a trivial example which illustrates these principles, consider a network of high school classmates. What would explain the pattern of friendship evolution measured over multiple measurements in a year? For instance, Mike and Jack might realize that they both have multiple common interests – both like tennis, Billy Joel, and Dungeons and Dragons. These common interests may lead Mike to form a friendship with Jack over the course of a year. Such an evolution would demonstrate homophily in action, as the common interest encourages the pair to hang out with each other after school. However, the fact that this friendship evolved, and was not present immediately, is a testament to transactive memory. At the start of the school year, Mike may not know about these shared interests because he might not know Jack, a transfer student. Over time, as Mike builds up a richer transactive memory representation of Jack's knowledge, Mike may begin to see that the two have much in common. This growth in transactive memory will feed into the evolving homophily process and likely lead to more frequent interaction. However, the budding friendship will be fundamentally embedded in a system where distant actions will

have strong local effects. For instance, the fact that Jack's family moved into the school district may be due to his dad's new promotion – a factor that is independent of any attributes of either high school freshman. This promotion may have been due to the fact that his dad gave a wonderful presentation, which led the company to start a new office in the neighborhood … and so forth. What is important to acknowledge, though, is the fact that there may be a hundred kids in the city with attributes like Jack's, but the fact that Jack showed up in Mike's district and thus is available to make the tie is due to millions of complex and highly non-local actions. Thus, an evolving friendship between Mike and Jack must be appreciated as a rich and complex process.

Each of the principles underlying Construct has been examined by a variety of researchers in diverse fields. For instance, homophily has been widely recognized as an organizing principle for human societies for a substantial period of time (e.g., McPherson, Smith-Lovin et al. 2001). Homophily may occur along a variety of dimensions and in a variety of settings, though most work with homophily has focused on socio-demographic dimensions. For instance, substantial research suggests that gender homophily suggests that boys are more likely to associate with boys in elementary and middle school. Homophily along multiple dimensions may be possible, since a pack of high school boys interacting with each other will be homophilus according to age and education as well as to gender. However, homophily need not be limited to socio-demographic attributes. Instead, homophily can occur due to common interests, shared experiences, and other "deep level" attributes (Harrison, Price et al. 1998). Indeed, as groups of individuals evolve, they often move from interactions based on surface characteristics to interactions based on more fundamental, core similarities which may not easily be immediately observable. While it is important to acknowledge that certain situations a social network may evolve due to heterophily (the attraction between two very different types) such as might be expected among men and women at a bar, such special cases are more often the exception rather than the rule and can often be explained by a deeper similarity between initially different groups (in this case, perhaps, homophily along the "looking for a date" dimension). Due to the principle of homophily, then, one would expect a social network to evolve in such a way that individuals with common attributes, knowledge, beliefs, or other salient features would be more likely to interact in future time periods.

In our discussing homophily, we have already observed that there are multiple dimensions – some of which are not immediately obvious – along which homophily may occur. However, it is necessary to acknowledge that there are dimensions along which users do not realize they are homophilous: for instance, when individuals initially interact based on surface-level attributes, it is quite common for them not to perceive the deep-level attributes of their new partners others. This absence of deep-level knowledge harkens back to the idea of bounded rationality by which individuals take actions based on their perceptions as opposed to an objective truth known by an omniscient observer (Simon 1957). More recently, this phenomena has been expanded and studied as the principle of transactive memory: the notion that individuals interact and form a shared memory unit in such a way that one individual may know that other individuals are knowledgable about some topic even without direct knowledge themselves (Wegner 1986; Carley 1991). This transactive memory often takes a substantial amount of time to build up, but can fundamentally shape the interactions between people. Specifically, transactive memory can be used to represent knowledge than an agent does not know. For instance, a husband may not be good with phone numbers but may remember that his wife knows them. Transactive memory can also be used to represent the fact that knowledge can go "out of date" – a one can remember that a friend was working for a large firm even though she was fired a year previously. These transactive memory perceptions are what can lead to fundamentally different types of interactions than would be expected if individuals were omniscient: if an omniscient salesman wanted to make a sale to the previously-mentioned large firm, he would probably not want contact the friend to facilitate the sale (and thus no link should be observed in a network). In the presence of imperfect information and potentially flawed transactive memory, however, such links can and will occur in an all-too-human network.

A final effect that has been explored primarily by researchers examining emergent and chaotic systems is that of the importance of small, non-local effects (e.g., Gladwell 2000). Such findings have suggested that a large event may be due to the confluence of a number of smaller events, none of which individually could have caused a specific outcome but which in aggregate can lead to substantial changes. Other researchers have begun to examine the statistical properties of large systems in order to understand how emergent behavior of small factors can combine to lead to a complex system (e.g, Epstein and Axtell 1999). Such research has suggested that some large, macroscopic behaviors may be due to the aggregations of many small changes over time, but it has also indicated that while the aggregate behavior of a system may be able to be predicted with some confidence the confident prediction of individual behavior may not be possible. Thus, any simulation which attempts to use a number of rules to forecast the behavior of a system must contend with the fact that predicting future behavior is, in a word, *hard*! While it may be possible to use homophily and transactive memory to guide how agents will interact in order to forecast network evolution, it must always be recognized that seemingly "random" effects may lead a network to evolve in a certain direction and can have profound effects upon overall outcomes.

While the above principles have been separately explored by researchers for a number of years, Construct is among the first generation of tools which has sought to unite the three together to study network evolution. This process of unioning these ideas, however, has been understandably complex. Therefore, it has been necessary to validate the model by comparing simulation results with case studies that have occurred in the real world. While the validation of Construct is an ongoing process, as the tool is constantly being improved, several validation case studies are worth mentioning. For instance, the original interaction model was validated using interaction data from Kapferer's tailor shop in Zambia (Kapferer 1972; Carley 1991). Construct has also been used to understand how information diffusion occurs in the presence of "smart agents", such as databases (Carley 1999) as well as on educational interventions such as web pages (Carley, Martin et al. 2009). Other work with Construct has examined the tiering effects that occur in social networks, such as occur when individuals are close to a small group of people but relatively distant to a larger group, and have found correspondence between simulated societies and real ones (Zhou, Sornette et al. 2005; Hirshman and St. Charles 2009).

It is important to note that other authors have identified alternative techniques for understanding evolving networks, including techniques that emphasize the transitivity and reciprocity of network relationships, or techniques that take into account specific network properties of the networks in question. For instance, models such as Sienna (Steglich, Snijders et al. 2006) choose to weight a variety of dimensions in determining network evolution. Construct is specifically an agent-based model, which allows individual actors to store knowledge, interact, learn, and modify their behavior in order to evolve the network. The Construct model presented in this chapter represents one of multiple ways in which researchers have begun to explore the evolution of networks. The three dimensions presented above have been proven in a number of settings and have been employed to some degree in many network forecasting models, sometimes as part of agent-based models and sometimes as parts of a different representation.

The tool

The graphical user interface for Construct , ORA's Near Term Analysis tool (Moon and Carley 2007), will allow a beginning user to harness some of the simulation power available to Construct and thus can serve as a gentle introduction to network evolution. The Near Term Analysis tool employs the Construct simulation engine in order to forecast the evolution of a network according to known social and psychological principles. The Near Term Analysis tool is a graphical user interface for interacting with Construct and can greatly assist the process of experiment setup, analysis, and debugging. The Near Term Analysis has been a part of the ORA package since <WHEN> in <YEAR>.

1. The data (1 sentence)
    a. Use the book's standard dataset

2. Topics (12-16 pages)
    a. DNA Measures (6-8 pages)
        i. Relative similarity
        ii. Relative expertise

Agents are decision-makers with varying information processing, socio-demographic, and access constraints and as such may or may not be human (Carley, 2002). Within Construct, agents go about their business interacting, communicating and learning each time period, as described in Figure 1. As agents learn or acquire information, they may change their preferred interaction partners and modify what they are likely to communicate. These factors, in turn, influence what types of decisions are made by each agent. A variety of factors influence who agents select as interaction partners, what they communicate with that partner, how much and how they communicate, whether they learn anything from that partner, and the accuracy and sustainability of that learning. Such factors include the agent's socio-demographic characteristics, information processing characteristics, proximity, and current position in the social and knowledge networks. The agent model has been described in depth in other venues (e.g., Carley, 1990 & 1992; Hirshman, Carley & Kowalchuk, 2007a & 2007b); thus, we concentrate here on both a high level description and details of those components used for the simulations reported.

Within Construct, agents both influence and are influenced by others. Agents who have influence over others can use that influence to escalate or de-escalate activity at a societal level by communicating information and/or beliefs. Social influence – as derives from shared attributes such as socio-demographic factors, shared knowledge, beliefs, and proximity – co-evolves with the spread of knowledge and beliefs (Carley, 1991). Consequently, in more heterogeneous populations where the lines of differentiation line up the chance of self- reinforcing beliefs at the group level is greater (Blau, 1977). Factors that are not influenced by the diffusion of information and beliefs include the agent's socio-demographic role (e.g., age, race, gender, level of education), the agent's basic cognitive limitations and information processing capabilities (e.g., likelihood of forgetting, risk taking, amount of information and beliefs that can be communicated or processed, and whether the agent has transactive memory), the size of their sphere of influence (at least in the short term), and factors that have resulted from socio-cognitive interactions (e.g., literacy, access to newspapers, radio and the internet).

Within Construct, agents develop likelihoods of interacting with others based on relative similarity (RS) and relative expertise (RE) (Carley, Lee & Krackhardt, 2001; Hirshman, Carley & Kowalchuck, 2007a). Relative similarity is a homophilly based mechanism (McPherson and Smith-Lovin 1987; Carley 1991) and derives from the idea that individuals are more likely to interact if they have more in common. Homophilly based interaction is a multi-causal phenomenon due to ease of communication, shared understandings, and comfort. The relative similarity of i and j, from i's perspective, is characterized as

$$RS_{ij} = \frac{\sum_{k<K}(AK_{ik} * AK_{jk})}{\sum_{j<I}\sum_{k<K}(AK_{ik} * AK_{jk})}$$

where individual i's relative similarity to j, is determined in terms of socio-demographics, knowledge, and belief items K in the agent-to-knowledge matrix AK.

Of important note: an individual is most relatively similar to itself, and each period will have a reasonably high probability of choosing to "interact with itself" and to avoid communicating with others. Just because an agent has the highest relative similarity with itself, however, does not mean that an agent will always interact with itself; indeed, due to the large number of other agents in the simulation, such avoidance of communication is relatively rare.

Relative expertise is a search based mechanism and derives from the idea that individuals are more likely to interact if one has information that the other wants. The relative expertise of j as judged by i is characterized as

$$if\ AK_{ik} = 0, then\ X_{jk} = AK_{jk}\ else\ X_{jk} = 0$$

$$RE_{ij} = \frac{\sum_{k<K} X_{jk}}{\sum_{j<I} \sum_{k<K} X_{jk}}$$

where individual i's relative similarity to j, is determined in terms of socio-demographics, knowledge, and belief items K in the agent-to-knowledge matrix AK (Schreiber 2006).

Agents are more likely to initiate interaction with another if they think the other has information they need and/or they are similar to them. However, there is a curvilinear relation between this familiarity and expertise; to wit, as agents initially increase in similarity (homophily) they are more likely to realize the other has expertise they need but as they increase still further in similarity they realize that the other is so similar there is no specialized expertise.

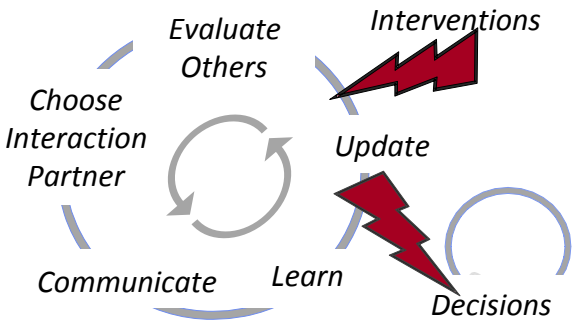
The researcher needs to specify the strength of each of these factors for agent-agent interaction. Herein, we set all human agents to use both logics and to at any time create a combined probability of interaction that is based on 60% similarity and 40% expertise. In both cases, individuals are giving and receiving information and the overall tendency to give versus receive is about 60/40 as identified by Valente, Poppe and Merritt (1996).

When setting up a virtual experiment in Construct the researcher needs to specify multiple parameters for each agent. This is often facilitated by the used of agent classes to parameterize multiple agents simultaneously. We next discuss: the number of agents in each of the classes of this experiment (section **Error! Reference source not found.**), the distribution of socio-demographic parameters for the agents of that class (**Error! Reference source not found.**), the distribution of cognitive factors for each class (**Error! Reference source not found.**), the sphere of influence for that class (**Error! Reference source not found.**), and the access constraints for that class (**Error! Reference source not found.**). While the full Construct model has a number of features that can be varied, such aspects were held constant for this simulation (Hirshman, Carley & Kowalchuck, 2007a).

Knowledge similarity and knowledge expertise, diagramed in **Error! Reference source not found.**, are fundamental to Construct's operation (Hirshman and Carley). Both similarity and expertise calculations rely on an agent's transactive memory and are fundamentally cognitive in nature. For instance, if an agent knows a fact, and the ego knows that the potential alter agent also knows it, then the fact contributes to a perceived similarity between ego and alter. In actuality, the alter may or may not know the stylized fact, or could possibly have known the fact and forgotten it. However, the ego agent perceives the similarity, and the perception leads to increased likelihood of interaction between the two agents. If the ego agent does not know the fact but perceives that the alter agent knows the fact, then the ego considers the alter to be a "relative expert" in that area, which increases the ego's probability to interact slightly (Hirshman and Carley). Note that if the ego knows that the alter does not know the fact, or if the ego has no transactive memory of the alter's knowledge, then the probability of interaction will be unaffected.

The algorithm outlined in **Error! Reference source not found.** is run for all facts for one ego/alter pair of agents, resulting in an overall score for that potential interaction partner. The agent then chooses to interact with an agent based on this probability score, an action that is fundamentally social in nature. It should be noted, however, that increasing similarity or expertise may not lead to an increase in interaction between two agents. If an ego agent learns a fact from an alter, the two agents will have increased similarity on an absolute scale; however, the same fact may cause the ego agent to become more similar to many other agents, thus leading to a decrease in relative similarity. Thus Construct, like human behavior, is highly non-linear and agent-agent interactions are as dependent on the similarity between agents as they are the differences between agents and other potential interaction partners.

Figure 68: The Construct Interaction Cycle



b. Construct Cycle (6-8 pages)

As mentioned previously, Construct is an agent-based model. This means that individual actors in Construct will individually evaluate their options and make decisions. For the purposes of this chapter, the decisions will be of the kind "who should I interact with?" although agents in Construct are able to perform much more complicated decisions based on their knowledge. As agents interact in Construct, the network on which the agents operate will gradually evolve. As agents choose new interaction partners, interact more frequently with some old partners, and drop others, the network itself will evolve. When this process is evaluated over all the agents in the network, profound shifts in network behavior can be observed.

In order to understand how Construct goes from the theory (described at the beginning of this chapter) and the calculations of relative similarity and expertise (described in the previous section) to changes in the overall structure of the network, it is necessary to understand how agents in Construct interact. While we present an overview of the five-step Construct interaction cycle in this section, we abridge it slightly so as to only focus on the outcomes which directly influence network evolution. Additional details about Construct, as well as information about its full capabilities, can be found in several technical reports about the tool (Hirshman and Carley 2007; Hirshman and Carley 2007; Hirshman and Carley 2008) or the project website, http://www.casos.cs.cmu.edu/projects/construct/.

Briefly, the Construct decision cycle is as follows. Each interaction cycle consists of the five sub-processes described in Figure 68, as read counter-clockwise starting from the top:

4. Agents evaluate their potential interaction partner using the relative similarity and relative expertise factors described earlier.
5. Each agent select an available agent with whom to interact using the scores derived from the potential partner evaluations.
6. Every agent communicates with its selected interaction partner; both the interaction initiator and the receiver send a message to the other.
7. Both agents have the potential to learn new information from this communication.
8. Agents update their knowledge and perform any actions specified in the simulation. This ends the time period, and a new time period starts.

It is worthwhile to delve into this cycle in slightly more detail in order to understand where homophily, transactive memory, and non-local effects can affect the simulation.

At the beginning of each time period, agents first employ their transactive memory to rank their possible interaction partners according to homophily-based similarity. Agents do this by creating a "probability of interaction" with each potential partner, a relative score which reflects the chance that any

Figure 69: Interactions in Construct

<DIAGRAM FORTHCOMING>

particular agent will be chosen (Hirshman and Carley 2007). This probability of interaction is affected by multiple factors: knowledge similarity and expertise, as described earlier, which are calculated using homophily and transactive memory; socio-demographic similarity (Harrison, Price et al. 1998; McPherson, Smith-Lovin et al. 2001; Hirshman, Martin et al. 2008), physical proximity, which is similarity due to similar positions in the physical environment (i.e., Barnlund and Harland 1963; Butts and Carley 2000), and social proximity, a catch-all for other types of similarity not explicitly modeled. While knowledge similarity and knowledge expertise must be present in all Construct simulations, the other forms of similarity are optional and can be enabled by the experiment designer. In some cases, such forms of similarity may prove to be extremely important; in others, it may be simpler to omit the factor and run the simulation based purely using knowledge as the driving factors. Regardless of which factors are used, the end result is effectively a score, for instance that an agent A will be twice as high a probability of interaction with agent B as with agent C, but three times as high a probability with agent D. This example is illustrated in Figure 69, and will be used as an example for the remainder of this explanation.

<EXENDS TO HERE>

Once agents have computed these rankings, each agent chooses its interaction partner or partners. Since there are usually a large number of agents in the simulation, it is not fair (or realistic) for one agent to choose first every time period. Thus, the order of partner selection is randomized. This means that while the interaction order may be A, B, C, D one period, it may by B, D, A, C the next and a still different order in a subsequent period. This interaction order can have subtle effects on partner selection. Consider the example from Figure 69, where agent A has the highest probability of interaction with agent D. If agent D is busy (perhaps already has been selected by another agent as an interaction partner), then agent A has to choose between agent B and C, and will likely choose the former since it has the higher probability. However, if A has the opportunity to choose first, it will likely want to select D. Note that the availability of the highest-ranked partner will not guarantee its selection: agents will use the probabilities of interaction in order to weight partner selection. Due to the randomness introduced in the process of interaction order and partner selection, Construct inherently introduces non-local effects into the simulation process: by making another agent select D as a partner before agent A has a chance to go, we can remove agent D from the list of A's possible interaction partners and thus affect the evolution of the network in that time period and later ones.

The partner selection process continues until all agents have selected available partners or choose to "interact with themselves" if no partners are available. Once all agents have partners, both the interaction initiator – the one who chose the interaction partner – and the interaction receiver – the one who was chosen – prepare a message to send to the other partner. Note that two messages are prepared, as communication in Construct is a two-way street. Both messages are drawn from the sender agent's current knowledge. These messages consist of a sequence of facts, beliefs, or transactive memory items regarding third parties. Thus, depending on simulation parameters, Construct agents have the potential to send informative statements ("a knight on the rim is dim"), beliefs on issues ("I believe that you should move your queen's pawn first"), and transactive memory about others ("agent B from Figure 69 is the local chess expert"). While the exact composition of the message is subject to cognitive limitations as well as item weighting (see, for instance, Hirshman and Carley 2007), the message sent will always be a subset of what the agent actually knows. The type of message transmitted represents another area in which random and potentially non-local effects can be introduced to the system, since agents can re-transmit facts learned from third parties. Thus, the very nature of the message sent by the agent can fundamentally affect interactions of distant third parties many time periods after the initial message has been sent.

Once both agents have prepared messages, the messages are sent from one agent to the other. At this point, agents have the opportunity to learn new information about their communication partner. For instance, if the other agent sent a fact, the receiver will have the opportunity to learn the fact – but then will also be able to create the transactive memory which states "agent C from Figure 69 sent me this fact, therefore he knows the fact" as well. Note that some agent cognitive limitations (e.g, Hirshman and Carley 2008) may affect the message elements learned by the recipients and can have profound effects on what agents actually perceive about their environment. Regardless of what the agent actually learns, the internal state of the agent will change slightly. This new knowledge may lead the agent to change its probability of interaction in future periods, as it realizes that it is relatively more similar to one agent than another. The knowledge may even make the agent relatively less similar to its old interaction partner, as the facts or beliefs transmitted may decrease the probability of interaction between them (or greatly

increase the probability of interaction between the agent and multiple other third parties, leading to a decrease in the original agent-partner relative probability of interaction). In this way, the learning is fundamentally intertwined with the evolution of the system as the changing knowledge drives the change in the observed network among agents.

At this point in the Construct cycle, agent interaction is complete. What remains is to gather statistics and update the simulation, as well as to allow agents to perform decisions (e.g., Hirshman, Carley et al. 2009). For the purposes of studying the evolution of an agent-to-agent social network, it is necessary to figure out who interacted with whom in order to store that information for subsequent analysis. However, it may also be useful to store who knows what information, an agent-to-knowledge network, in order to better understand why two agents were drawn to each other or to understand the rate at which knowledge similarity is increasing over time. The Near Term Analysis tool provides a snapshot of many of these networks in order to facilitate analysis and to help the beginning user understand how networks are changing over time. Once these statistics have been gathered, the next simulation period starts and the network evolution continues.

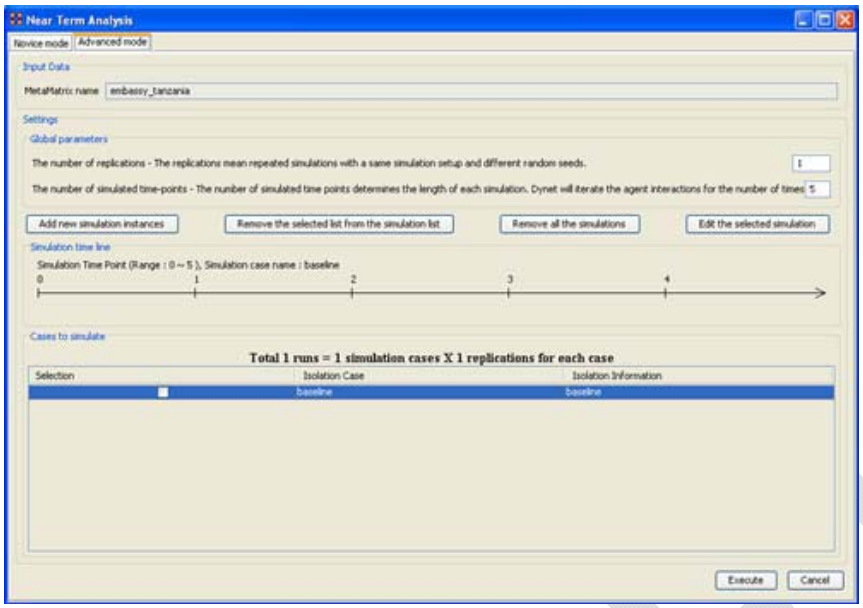
c. Attributes
d. According to relationships/links

3. Walkthrough Near-term impact report (6-8 pages)
    a. Show how to set up experiment
        i. Set up experiment parameters
            1. Indicate the number of replications
            2. Indicate the number of time periods

We can examine the impact of removing a particular entity from the network by using Near Term Analysis (NTA). NTA removes a given entity and estimates the changes in the remaining network.

This provides a what if?... analysis of the network. We can simulate strategic interventions or eliminations of certain agents, and examine how the network should react and change. For example, we can look at how the Tanzania Embassy bombing network changes if Ahmed Ghailani and/or Wadih El-Hage are eliminated.

To run a Near Term Analysis:
1. Start the Near Term Analysis window
• Return to the ORA interface.
• Make sure the meta matrix is highlighted in Panel 1. If it is not highlighted, single-click on it.
• Go to Analysis in the menu bar and select Near-Term Analysis. A window titled Near Term Analysis will pop up.
NTA has two modes, Novice and Advanced, displayed as tabs in the Near Term Analysis pop-up window. Novice mode provides an automatic analysis set-up, but does not allow users to set up their own hypotheses.
• Click on the Advanced mode tab to switch to Advanced mode.
Below is a screen capture showing the Advanced mode tab in the Near Term Analysis pop-up window:



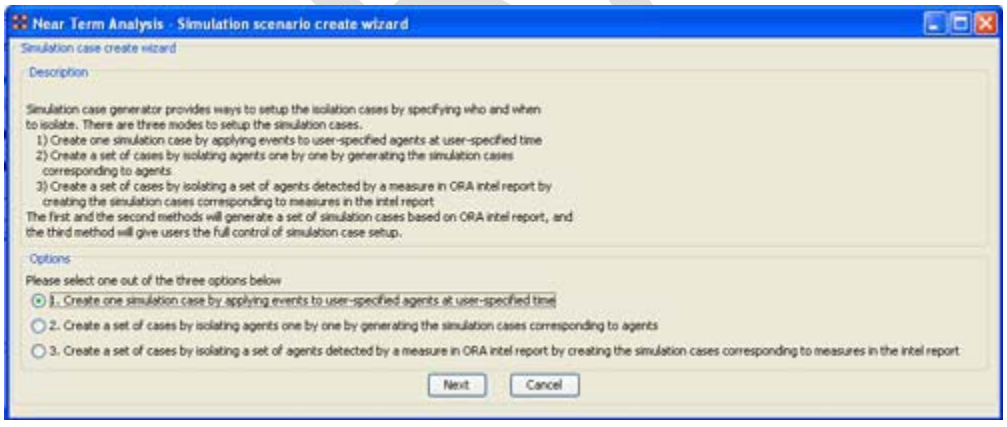
In the field titled Settings, find the number boxes on the far right. Change the number of replications (the top number box) to 2, and the number of simulated time-points (the bottom number box) to 52. The timeline in the Simulated time line field will expand to include 52 time-steps.

2. Open the simulation scenario wizard

• In the middle of the pop-up window, find and click the Add new simulation instances button. A small window titled Near Term Analysis - Simulation scenario create wizard will pop up.

Below is a screen capture showing this wizard window:

•

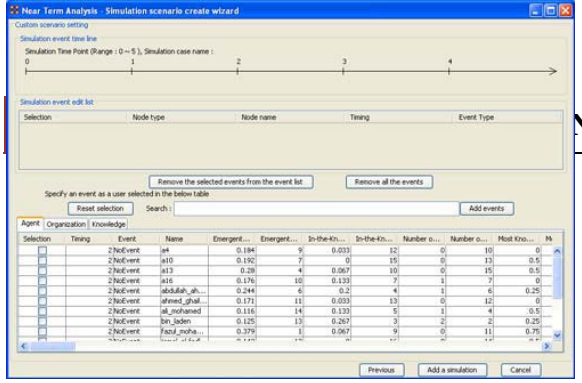


In this wizard window, make sure the first option in the Options field is selected.

Click the Next button. The wizard window will display options for creating a custom scenario.

59

Below is a screen capture showing this custom scenario field in the wizard window:



59

Below is a screen capture showing this custom scenario field in the wizard window:

3. Remove Wadih El-Hage

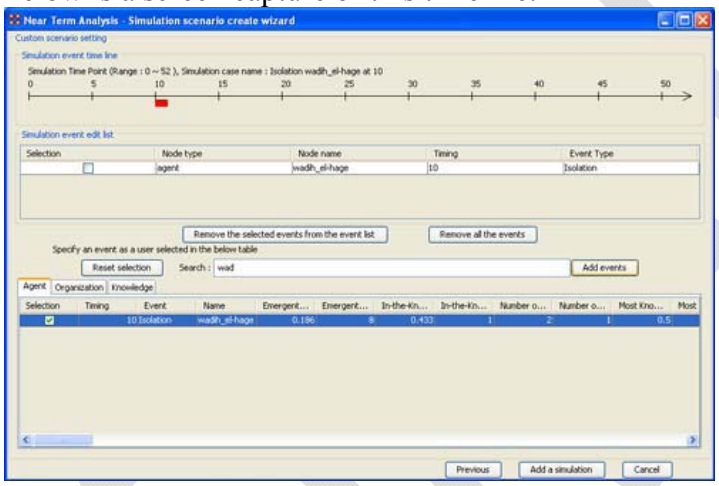
• Click the box for Wadih El-Hage in the table of agents in the wizard window. Change the Timing box to 10 in the same row.

• To add this event set-up to the Near Term Analysis, click the Add events button located next to the Search field (above the list of agents).

The timeline at the top of the wizard window will mark the isolation of Wadih El-Hage at time 10.

60

Below is a screen capture of this timeline:



Click the Add a simulation button at the bottom of the wizard window. The wizard window will close automatically. In the NTA window, the new simulation case will appear in the Cases to simulate field.

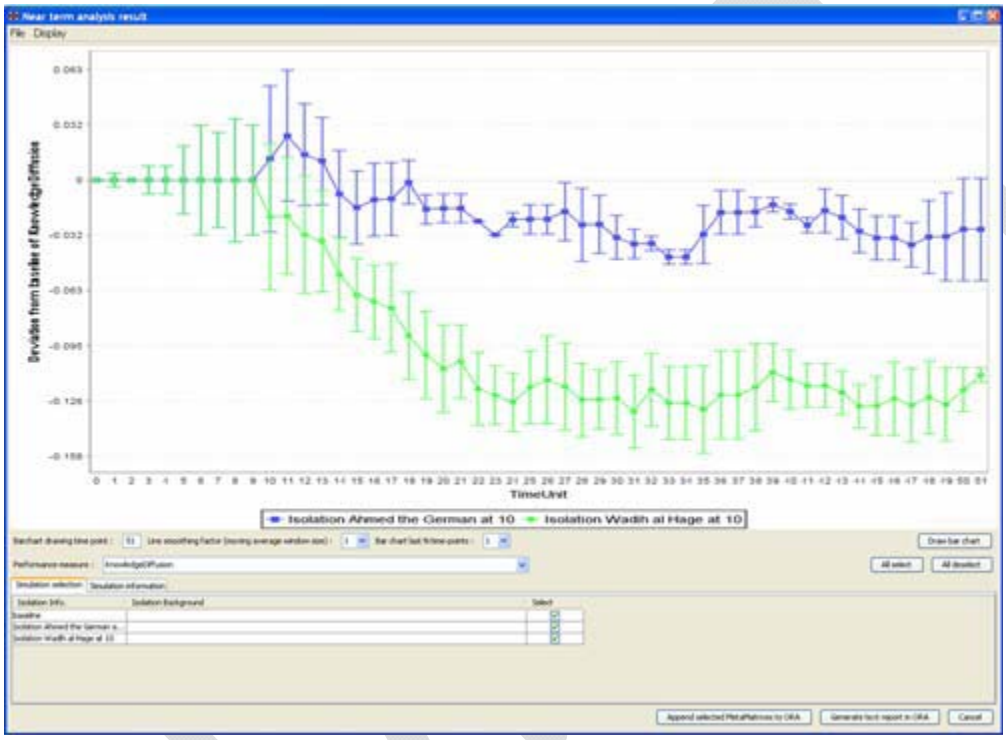
4. Remove Ahmed Ghailani

• Follow Step 3, but replace Wadih El-Hage with Ahmed Ghailani.

Your two simulation cases appear in the Near Term Analysis window in the Cases to simulate field, along with the Baseline case. The Baseline case is simply the network with no entities removed.

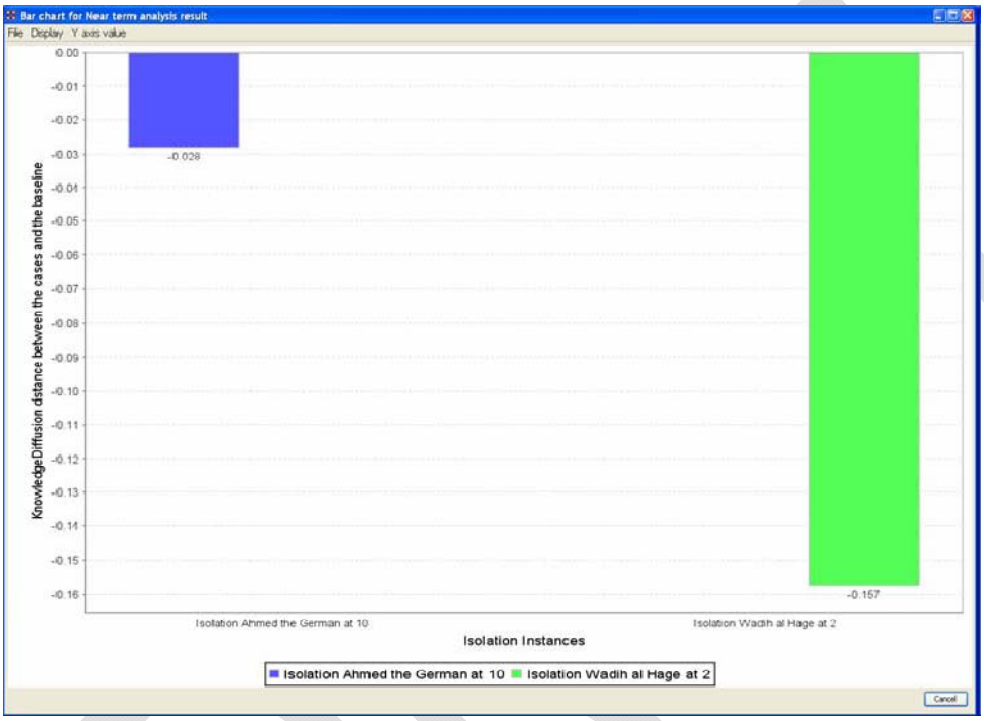
You can view your different simulation cases by clicking on them in the Cases to

simulate field, and the simulation time-line will change accordingly.
5. Run the Near Term Analysis
• Click the Execution button at the bottom of the Near Term Analysis window.
• A small window will pop up to warn you that the execution may take a long time. Click OK. Another small window will pop up to show you how the simulations are progressing in the Near Term Analysis.
6. View the results of the Near Term Analysis
• When the analysis is complete, a new window titled Near Term Analysis Results will pop up.
61
Below is a screen capture of the results pop-up window:



These two performance lines correspond to deviations from the baseline of knowledge diffusion over time. Examining this performance change over time reveals how much an agent's elimination impacts the performance of the organization.
• To draw a bar chart, click the Draw bar chart button in the results window. The

resulting bar chart displays a performance comparison for each time-point.
62
Below is a screen capture of the bar chart:

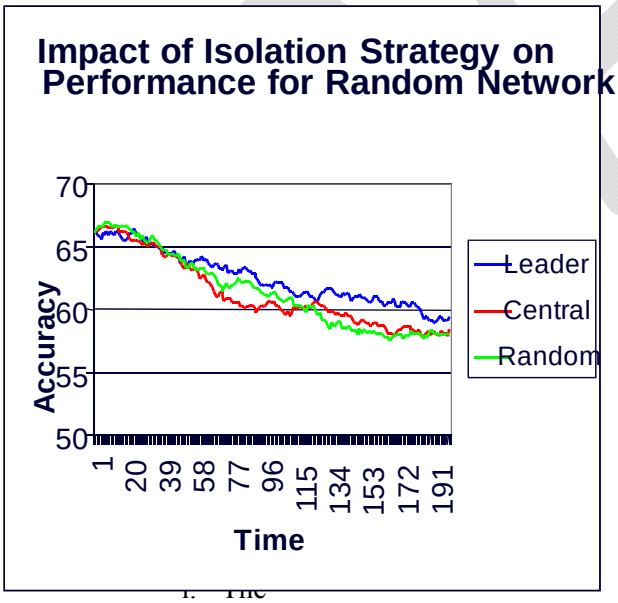


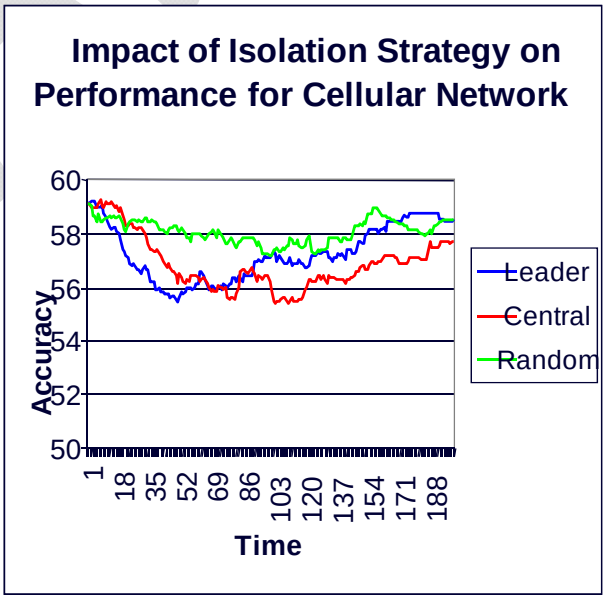
The results of the Near Term Analysis are also available in HTML format.
7. Save the results of the Near Term Analysis in HTML format
• Go to File in the menu bar and select Save the html report. A file chooser window will pop up. Navigate to the location you want and type a filename for your HTML report.
• A window titled Image Size Input will pop up. Type in the width of the image and click OK.
• Another window titled Project Saved will pop up displaying the location of your saved analysis results. Click OK.
8. Save the organizational structures resulting from your simulations in ORA
• Return to the ORA interface.
• To save the Agent-to-Agent communication network at time-step 40 in the

simulation, go to File and select Save the Agent-to-Agent communication matrices during evolution.
• A small window titled Save the probability of interaction matrix will pop up. Click the Yes button.
• Another small window titled Save timing will pop up. Enter 40 and click OK.
• A final window titled Saved will pop up. Click OK.
63
You can view saved simulations in Panel 1 of the ORA interface.

ii. Set up the baseline and scenarios
    1. Baseline is automatically included in the initial set up
    2. Set up other scenarios
        a. Indicate which node(s) and when the isolation(s) should occur

b. Show how to read output charts
    i. The output charts indicate the performance and the knowledge diffusion over time for each of the scenarios and the baseline





Problem Set

d. What 5 factors are considered when Construct selects an interaction partner for an actor?
e. What is the minimum number of alters a specific agent can interact with in a given time period? What is the maximum?
f. What are the four types of data-t items that an actor can communicate to another actor?

g. Does an actor maintain transactive memory about himself?
h. Why does an analyst run more than one replication for an experiment?
i. What are the four node classes that represent an organization in construct?
j. In construct how is organization performance determined?
k. What does the measure knowledge diffusion indicate?
l. Can the construct script file that the near-term impact report creates, be used in the construct stand-alone software?
m. What is relative similarity?
n. In the dataset which actors/dyads have the greatest level of relative similarity (NEED TO GET ORA TO COMPUTE THIS???; maybe use the existing redundancy measures???)
o. What is relative expertise?
p. In the dataset which actors/dyads have the greatest level of relative expertise (NEED TO GET ORA TO COMPUTE THIS??? maybe use the existing exclusivity measures???)
q. What is the purpose of running a baseline scenario in the near-term impact report?
r. #1: isolate an actor
    i. Run the near-term impact report for a 100 period (10 iterations) and isolate Ahmad the German at time period 20. How does the isolation impact the organizational knowledge diffusion before the intervention, then after the intervention? How does the isolation impact the organization performance, before and after?
s. #2: Isolate a knowledge node
    i. Run the near-term impact report for a 100 period (10 iterations) and isolate marketing knowledge at time period 20. How does the isolation impact the organizational knowledge diffusion before the intervention, then after the intervention? How does the isolation impact the organization performance, before and after?
t. #3: Isolate two entity types
    i. Run the near-term impact report for a 100 period (10 iterations) and isolate both Ahmad the German and marketing knowledge at time period 20. How do the isolations impact the organizational knowledge diffusion before the intervention, then after the intervention? How does the isolation impact the organization performance, before and after?
u. #4: Exploring options
    i. Which if the following interventions are expected to have a greater reduction in organization performance at time period 85?
        1. A) Isolate ahmed the German at time 20
        2. B) Isolate ahmed the German at time period 40
        3. C) isolate Osama bin laden at time period 20
        4. D) isolate Osama bin laden at time period 40
        5. E) Do nothing

CHAPTER 11: Getting Data – Building the Julius Caesar MetaNetwork

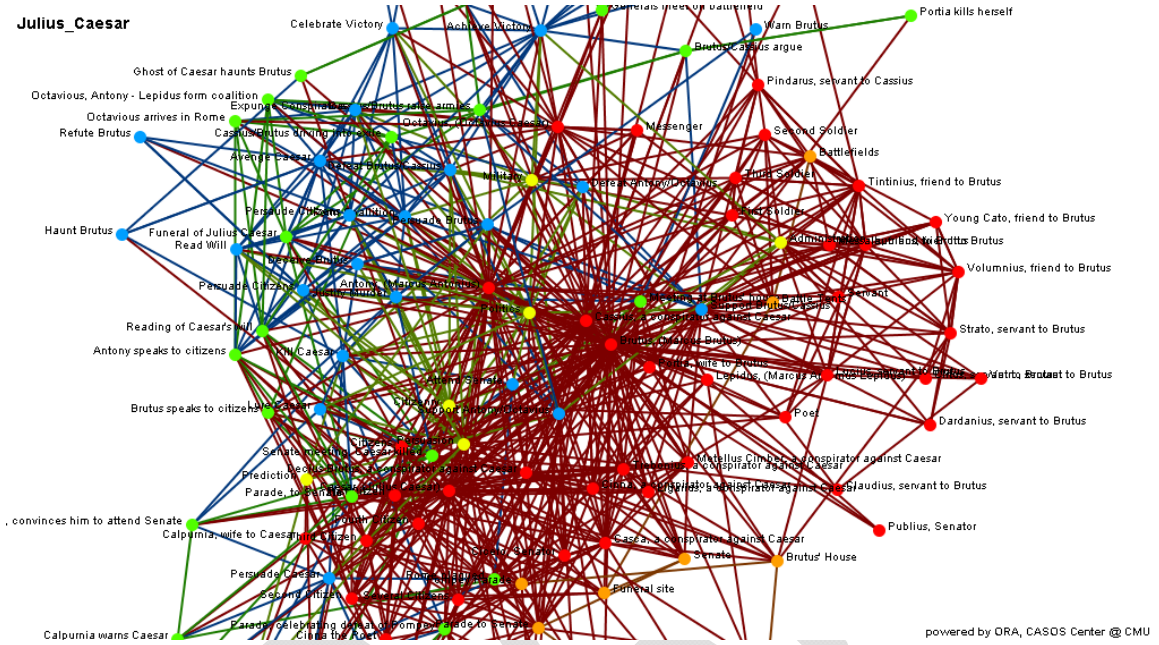


Figure 70: Partial image of full Julius Caesar MetaNetwork Visualization

So how did we get here?

Imagine a biologist that needs to know the details of what a butterfly looks like. Well, it is understood and should be understood; he or she had better be good at catching butterflies. You would not give the microscope to someone that couldn't catch a butterfly and with instructions to analyze in detail the intricacies of what the microscope reveals. Hence, it follows then that the most value that can be taken from Dynamic Network Analysis is to put it, the microscope so to speak, in the hands a capable network analyst skilled at sizing up networks, catching butterflies so to speak. It is that spirit; we will discuss the building of the Julius Caesar network and all that went into making key decisions about our model, which ultimately affect what we can take from our model in terms of analysis. Put first, a word about what Dynamic Network Analysis is capable of doing for the network analyst.

Dynamic Network Analysis is a powerful tool that can tell us amazing insights about network data that is of interest to us. That does not mean that we can simply feed a bunch of random network data into the machine of Dynamic Network Analysis, process our algorithms, and derive value from the results, hoping that the algorithms will sort it all out. No, I am reminded of the old computer programming dictum: garbage in equals garbage out. In many ways, Dynamic Network Analysis is the same way. We need to put careful thought into what it is that we want to model and what it is we hope to learn from that very same model. Therein we hit upon a key to getting the most out of Dynamic Network Analysis and that is putting the techniques and the science of DNA in the hands of those that have distinguished ability to size up a network, an innate ability to see what constitutes a meaningful network attribute on a micro scale. Because, such a person can wield the techniques of DNA, much like a microscope in the hands of a

biologist learned in the way of catching butterflies. Just like such a person, the DNA scientist can look at your network data and draw all sorts of insights heretofore unavailable in the traditional disciplines of link analysis. But, it must be put in the hands of a person skilled at network analysis.

To borrow another metaphor: imagine a police officer skilled in the art of using a hand gun. You might just do well with a standard issue 38. However, you couldn't necessarily give him the most advanced weapon out there if he is unaccustomed to firing the 38 accurately or at the very minimum with a modest amount of acumen. DNA is the same way. You need someone who can fire the old model of traditional link analysis with some degree of accuracy. DNA can do more in such a persons hands and the more skilled network analyst is just the type of scientist who will invariably get the most out of DNA.

With apologies to the great bard, let us now discuss how we built our Julius Caesar Meta Network and what problems we encountered along the way, what questions we had to resolve and what limitations we had to live with.

It is our hope that these experiences, which we will share, will be beneficial to you as you attempt to build a meaningful network model of whatever it is that you are interested in putting under the microscope of DNA. You can learn from the roadblocks we encountered and read about questions we had to ask ourselves as we constructed the Julius Caesar network from the ground up. Ours is a sufficiently complex network to work with for purposes of this book. Yours might be vastly more complex and greatly more ambitious. Nonetheless, you will encounter crossroads when deciding on how to handle your own network data to get the most out of it. We present to you then the crossroads we faced when building Julius Caesar from the first character on up to the last. The questions we faced are likely to be the same ones you will have to deal with at some point in building your own MetaNetwork.

We must also mention that the chief tool employed in the science of DNA is currently ORA, a network analytic toolkit developed at CMU-CASOS, AutoMap a network extraction tool developed at CASOS, and Construct a network evolution tool developed at CASOS.

More than just agents – thinking beyond Whosville

We all remember Dr. Seuss's famous story about the *Grinch That Stole Christmas*. In it, the town villagers lived in a place called *Whosville*. In all likelihood, the first step in building your Dynamic Network model is determining who will live in your Whosville. In our case, the principal actors of our Whosville is quite literally the actors since in Julius Caesar we are dealing principally with *Whos*, a cast of characters nicely laid out for us by William Shakespeare. However, as it applies to any network you are interested in analyzing, the most natural starting point is establishing the *Whos* that will live in your own network Whosville. It is our hope this is a relatively easy concept to grasp for the most uninitiated of network analysts. After all, most network models of any organization are really a *Who* structure – who reports to Who, Who does what, how often will Who do this, *Whos* on third? (To channel Laurel and Hardy).

Therefore, the first step is typically the easy step: determining what are your *Whos* and *Who* are they. After all, your *Whos* don't have to be actors in a play. They can be computers in a network, they can be terrorists in a cell, they can be basketball teams—collections of people; they can be concepts such as words in a visual word map. Your *Whos* can literally be anything but more often than not, they might be people. People seem to be the complex entity that most organizations are interested in studying for one reason or another. That is just fine, so long as you keep in mind that your *Who* can literally be anything and need not necessarily be a real person or people. A particular contagion can just as easily constitute a *Who*. Therefore, that leads us to the first step of the Julius Caesar network model and that is determining exactly *Who* is *Who* in our network model – who will live in our *Whoseville*. This was neatly spelled out for us by William Shakespeare. We had to look no further than the following cast of characters in the opening part of the play:

Julius Caesar Cast of Characters – Our Whos

Antony, (Marcus Antonius)
Artemidorus, of Cnidos, a teacher of rhetoric.
Brutus, (Marcus Brutus)
Caesar, (Julius Caesar)
Calpurnia, wife to Caesar
Casca, a conspirator against Caesar
Cassius, a conspirator against Caesar
Cicero, Senator
Cinna, a conspirator against Caesar
Cinna the Poet
Citizens
Claudius, servant to Brutus
Clitus, servant to Brutus
Dardanius, servant to Brutus
Decius Brutus, a conspirator against Caesar
First Citzen
First Commoner
First Soldier
Flavius, a tribune
Fourth Citizen
Lepidus, (Marcus Antonius Lepidus)
Ligarius, a conspirator against Caesar
Lucilius, friend to Brutus
Lucius, servant to Brutus
Marullus, a tribune
Messala, friend to Brutus
Messenger
Metellus Cimber, a conspirator against Caesar
Octavius, (Octavius Caesar)
Pindarus, servant to Cassius
Poet
Popilius, (Popilius Lena)
Portia, wife to Brutus
Publius, Senator
Second Citizen
Second Commoner
Second Soldier
Servant
Several Citizens
Soothsayer
Strato, servant to Brutus
Third Citizen
Third Soldier
Tintinius, friend to Brutus
Trebonius, a conspirator against Caesar
Varro, servant to Brutus
Volumnius, friend to Brutus
Young Cato, friend to Brutus

To Who or not to Who

Now, we get to our first crossroad. Even the Whos are not always so obvious. When we first looked at our handy list of *Whos*, we thought it would be relatively easy to simply compose a list of all the characters in the play and go with that. It still was relatively easy, but not without some consideration. You see, we had to ask ourselves if we were interested in Shakespeare's penchant for multitude of minor characters. We debated as to the best way of handling the minor characters: 1) we could ignore them and leave them out of our network model; 2) we could list them all as they are for better or worse, even if they don't ad much to our network model; or 3) combine the ones that were nearly identical in nature and fulfilled the same purpose into one node, that is one Who, to represent them all. This was the decision we were faced with when building our model. Here is how we handled it.

For instance, Shakespeare has several minor instances in the play where he refers to the following characters:

First Citizen
Second Citizen
Third Citizen
Citizens

As we mentioned we could simply call them all "Citizens" and then build our network with the "Citizens" in mind from that standpoint. We could also, just forgot about the citizens, dismiss them as minor characters but the astute network analyst and literary analyst for that matter, and could clearly discern for themselves that the Citizens— though comprised of small bit characters — serve a vital function and critical role in the plot of Julius Caesar.

You see, going back to the early parts of the play; we encounter Cassius who wants to desperately overthrow Julius Caesar. He can't do it alone because he does not have the political clout to pull it off, but he does have the ear of Brutus, a very powerful figure and friend of Julius Caesar. However, it is not the reason why, per se, that Cassius wanted to bring Brutus into the fold. Yes, he rightfully pegged him as pivotal to the success of his plot, but not merely, because he was powerful. Brutus, Cassius reasoned, could motivate the Citizens. Cassius couldn't do that nor did he believe the other plotters were capable of winning over the Citizens after they killed Julius Caesar. Therefore, by extension, we asked ourselves, if Cassius believes that Brutus is necessary to motivate the Citizens in the wake of a deposed Caesar, then the Citizens server a vital role in the play. Without them on the side of the plotters, Cassius rightfully foresaw doom. Brutus becomes all the more important because Cassius has identified him as having that rare quality to motivate the masses - that is the citizens. Cassius believes Brutus' charisma and the respect he commands from the people alone, will allow the plot to work. Moreover, it is Brutus that Cassius believes is critical to running a Caesar-less empire after his nefarious plot is carried out to its conclusion. Cassius was smart in his regard that the citizens were of crucial importance and Brutus even more so because he could mobilize them. One might even make the case that Cassius could have cared little for Brutus and his idealism, because he wanted Brutus for the task of motivating the Citizenry. The problem, as we learned for Cassius, is that though Brutus was good at motivating the Citizens, there was someone better: Marc Antony.

Curiously, Cassius seemed to want to nix him along with Julius Caesar but as you may recall, Brutus argued against killing Marc Antony because, to paraphrase him, they would be seen as butchers. Instead, he wanted to serve up Caesar as sort of ceremonious offering to the Gods because he "loved Rome more." One could imagine Cassasius thinking privately that Brutus is lost in a cloud of idealism. Ever the pragmatist, Cassius would just as easily have killed Antony but he sensed that Brutus would not go for it. Unfortunately, Cassius was probably right. After all, the plot ultimately failed because Antony lived to mobilize the Citizenry against Cassius and Brutus. However, we digress.

Therefore, to ignore the Citizens would not be a wise move if we wanted to gain a deep insight into the network that makes up the world of Julius Caesar as Shakespeare presented it. However, does it make sense to feed into our algorithms and model the minor differences contained within the group of Citizens such as First Citizen, Second Citizen, Third citizen and then finally Citizens in general? That answer is not so clear.

This is a point where you have to use your better network judgment.

To be thorough you might want to capture every bit of network data you can get your hands on and then look at it all anyway and see if it tells you something meaningful. On the other hand, you don't want your network model so convoluted as to be of dubious practical value because you have modeled the so-called obvious. So, what did we do? How did we resolve this question?

If you have been following along since the beginning of the book, you will see we captured everything. So we took the more thorough approach and included all the variations of what amounts to essentially a group that called be lumped together under Citizens. Our results seemed to show no significant difference in our network model if we would have eliminated the First, Second, Third Citizen all together. The node Citizens represented them all in most characteristics and there was not anything unique to be gleaned from observing the first hand ties that each First, Second, Third Citizen actually

held. That does not mean it was a good or bad decision. It just tells us something more that we may or may not be interested to learn more about down the road. We took the chance that it would.

So, what exactly looks different in our network model based on using each individual citizen and even the group of Citizens in general? Let's examine our visualization and see if we can discern any noticeably connections or lack thereof. For the sake of clarity, we will only deal with one graph, agent by agent, or a *Who* by *Who* graph in the Julius Caesar network and see what the result is going to be:

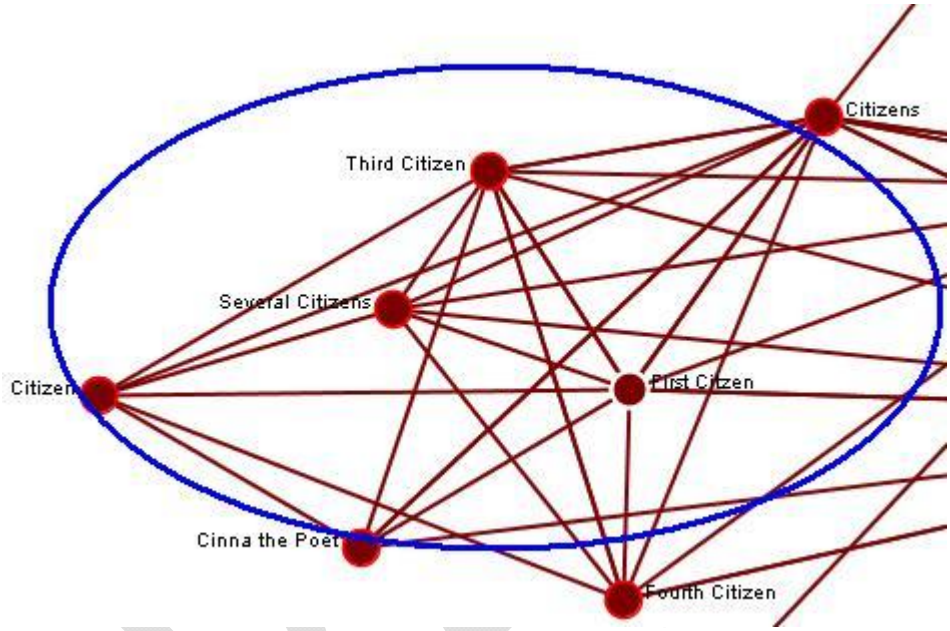


Figure 71: Detail of individual Citizen (Who) characters appearing in Julius Caesar. Is there something unique about any one as an individual versus all of them as a collective whole represented by the blue ellipse?

Below is the entity status for the node, the *Who*, called Citizens.

Figure 72: Citizens has 20 links. Image provided by ORA Visualizer.

Examining all the *Whos* that are called Citizens (e.g., First, Second, Third, etc.) we learn that each one has anyway from 16 – 20 connections much like the Who that contain Citizens as a whole. We can also observe that this whole part of our Julius Ceaseer MetaNetwork seems to be highly interconnected. It is almost like its own little corner of our network model. What does this tell us? What can we conclude from this?

We might reasonably conclude that we could indeed collapse the various instances of citizens into the Who id that is called Citizens. This would have eliminated this little interconnected group and produced one group called Citizens, which would be reasonably comfortable in stating that it accurately captured all the sub components of the First, Second and Third Citizen and therefore we could have eliminated them. This then was our first crossroad and how we dealt with it. Now on to our next.

A question of Knowledge versus Tasks

Another crossroad we faced in building our network model came down to deciding on whether to identify tasks, knowledge, or both, as we began to identify exactly what was the most important parts of the Julius Caesar network of interest to us. Before we get into the decision, we made and how we did it, a word about Knowledge and Tasks as they related to constructing MetaNetwork suitable for DNA.

By now we have become familiar with many common Measures that tell us very powerful insights into the structure of our MetaNetwork. Those Measures often factor in the Knowledge and Tasks component interchangeably. That is the Measure takes as input either an AR graph (Agent by Resource) or an AK graph (Agent by Knowledge). That is you can decide which is of utmost value in drawing from your network model for maximum insight. You can make a choice whether to input a Resource or a Task, because in many instances, a Resource can be a Task and so to can either of those be Knowledge under some circumstance. Let us think of a few examples.

Having a bachelor's degree in computer science can be Knowledge. That Bachelor's degree can surely be a resource for an IT Department. Such examples were evident in our model of Julius Caesar.

When we analyzed the cast of characters, locations, settings, tasks, we began to think about what resources were necessary for the plot to work as Cassius was hoping it would. We could come up with Military, Power, and Administration – maybe a dagger to kill Caesar? Weapons then could have been on our list. Then we started thinking about other resources, more physical in nature, the Senate chambers? Was that a resource? It could be. However, did a building have some kind of relevance for our network model? It did not appear to be the case.

Ultimately, we concluded that physical resources were not instrumental in our plot. Other than a dagger, which seemed perfunctory to the plot, there were not a whole lot of physical resources beyond the realm of obviousness, such as shoes, cloaks, daggers, laurels, that we really cared about modeling. The true resources that needed to be leveraged to carry out the plot were largely resources of the mind – they were knowledge based.

Brutus has the knowledge of persuasion, which was a critical resource in many ways. Antony too had the same resource and in greater value than Brutus because he was able to in the end convince the Citizens to support his cause. The soothsayer, Archimedoras, and the wife of Julius Caesar, Calpurnia, all had the resource of "prediction" – again, knowledge based. Therefore, we made the decision to focus on knowledge. Since many of the measures take as input an AR or AK graph, we decided to go with an AK (Agent by Knowledge) graph. Therefore, that was a decision that needed to be made about our network.

That is not to say you can't have both an AK and AR graph. It was just that in our case an AR graph wasn't needed. In constructing a detailed model of say Al Qaeda and the Tanzania Embassy bombing of 1996, you very well have both. You would have a Resource as a bomb material and have a Knowledge of bomb making. You could therefore model both such resources and knowledge sets differently. That was not necessary in Julius Caesar.

Planning your network for now or over time

The last major crossroad we faced in building our network model of Julius Caesar is really a question of using some of the more advanced features of dynamic network analysis and that is to model a network over time. Making such a decision early on in the network model planning stage is key to determining how you will go about collecting the necessary and appropriate MetaNetwork data to analyze over time. Here is why.

It is obvious that the plot of Julius Caesar is something that unfolded over time. In fact, it is all clearly delineated in five acts (I, II, III, IV, V) and it is easy to created and even visualize mentally a timeline with the Acts clearly denoted and all the events and characters plotted on the accordingly. However, in our first MetaNetwork we took the complete entire play of Julius Caesar as a whole from beginning to end. We modeled all the characters at once, not as they were introduced; we networked events as they had connections to each other, not as they appeared in time, we graphed our knowledge nodes as they tied to other agents, not on the basis when those agents appeared and when the knowledge node became evident. Therefore, when we began we created this massive MetaNetwork of complete data gleaned over one massive event called Julius Caesar and treated it as just that – one massive event. We didn't model it in accordance with how it evolved over the five acts. We model all five Acts together. So what you ask?

Well, there is a substantially different process to model a network as it evolves over time versus just taking all your data as one massive input. First off, you might be greatly interested to see what events seemed to shape the course of the ultimate Network Model and to do this sort of function; you need to model the network data as it evolves over points in time. You can't just take all information from all five acts and produce a MetaNetwork, which we did. Such a model still is highly insightful in terms of

structure but it tells you nothing about the networks evolution. In addition, why are we interested in evolution of our network model? Because if we can study how the network evolved to this point, we might know a little something about how it is evolving today, thus we might learn what the network will look like in the future. Moreover, we can run trials an experiments, using the most advanced features of DNA, and see what happens when we impact a network by removing a certain agent, knowledge, resource, or task along the way. We can mathematically show how the evolution of that network would likely be impacted. Now that is powerful stuff. Here is how you would do it.

Essentially, to capture network evolution you need to establish what time intervals you would like to capture. You would then create a MetaNetwork as we have learned how to in the early chapters, for just that one time interval. Then, depending on how many intervals you have established as being inherently interesting to you, you would build a MetaNetwork for that time interval based solely on the data available for that time interval. Once that is complete, you can then stack all these MetaNetworks together, metaphorically speaking, and see how change has propagated from one time interval to the next. Once again, The Organizational Risk Analyzer has the tools to perform a mathematical over time analysis on your network data capture over time. Now, how does this differ from how we built our first MetaNetwork? Not much, but there is more work to be done.

You would still make connections from one agent to another exactly as you did before except only with in the confines of that time interval. This is to say that if we wanted to conduct an overtime analysis of the Julius Caesar Network we would start by establishing exactly what time intervals we want to capture. For lack of a better time interval, we might opt to choose the five Acts themselves as the best time interval to study, if not because obviously the Shakespeare thought his play fit neatly into the confines of five major acts and we can conclude that there is obviously five major developments in each act. Therefore, we would expect to see some meaningful network change on an overtime analysis of each of Julius Caesar's five Acts. Therefore, we need to follow the steps of building a MetaNetwork exactly as we did before, but only limit it to what is happening in terms of agents, knowledge, tasks, etc. in that act and in that act only.

To illustrate this point a bit further, if we are planning an over-time analysis of Julius Caesar we would not use the full cast of characters as they were so nicely outlined at the beginning of the play for us, we would only use the agents that appeared in Act I. Therefore, if Octavius doesn't show up until Act III, then he would not exist anywhere at all in our MetaNetwork of Act I and II respectively. However, here is where it gets real interesting. What happens when someone dies?

You may recall that Julius Caesar is killed in Act II. Does that mean he is no longer in the MetaNetworks of Acts III, IV, and V? No. Once he is introduced, he is part of all the subsequent networks even though he is no longer alive – he just has the quality of being dead. In this case, that would be attribute. It may sound a little odd that if he died how he could still be in the network, but just think about it for a second. Even though he is dead, the wars that follow are all fought because of him in some sense, so how could he not be in the network? In addition, to make one last minor point, he does return later in the play as a ghost. The beauty of it is that DNA can handle that no problem. Traditional link analysis might have a difficult time on that one.

CHAPTER 12: Extracting Networks from Texts – Content Analysis

From Texts to Networks

Introduction

Texts can be an important source of data about networks. In the social sciences, text data are generally thought of as qualitative data. In computing, text data are often treated quantitatively. Cutting-edge text analysis techniques bring together these two research paradigms with the ultimate goal of providing users with effective and efficient insights into the structure and behavior of dynamic, complex, and socio-technical networks of any size (Bernard & Ryan, 1998; K.M. Carley, 1990). These methods can be suitable for testing and developing hypotheses and theories about real-world networks (Corman, Kuhn, McPhee, & Dooley, 2002; P. Monge & N. Contractor, 2003). Network Text Analysis (NTA) is a family of methods developed for this purpose (Danowski, 1993; van Cuilenburg, Kleinnijenhuis, & de Ridder, 1986). NTA supports the effective and efficient extraction of the structure, behavior and meaning of networks from text data (K.M. Carley, 1997).

The next three chapters teach you how to use natural language text data in the context of network analysis. The textual information that is relevant for answering a research question can be represented as unstructured or structured information. The next three chapters describe the basics of how to distill relevant information and different types of networks from texts, and what metrics you would use in analyzing the extracted information and data.

Objective

The goal with this chapter is to introduce you to the applicability and usage of text analysis in the domain of network analysis. You learn how to extract relevant information and structured data from text data in an efficient and systematic fashion, how to perform appropriate analysis on the extracted data, and how to interpret and evaluate your results. Secondary goals are to understand how the different choices that you make throughout this process impact your results. You gain practical, hands-on experience in working with the AutoMap toolkit for text mining and the ORA toolkit for social network analysis. More specifically, you learn how to import the data extracted with AutoMap into ORA, and how to use ORA to analyze these data. Both, AutoMap and ORA, have been developed by the CASOS Center at CMU, and have been applied by multiple groups from academia, government and industry across different domains and data sets. In summary, the learning goals are:

1. Information and Relation Extraction: gain theoretical, methodological and practical experience in distilling relevant information from texts. Understand and perform several natural language processing techniques.
2. Network Analysis: learn the basics as they apply to relations extracted from texts, get hands-on experience with network analysis software, generate and interpret network analytical results.

Background

Terminology

For the context of this book, text data does not include speech data. In the domain of language data analysis, text collections are also called *corpora*. Examples for corpora that can be collected include news coverage, books, legal documents, public debates, interviews, annual reports, and mission statements as well as social media such as emails, chats, blogs, wikis and webpages. Due to technical advancements, it has become fast, cheap, and easy to collect and store large amounts of mainly unstructured, natural language text data. The availability of large corpora has further perpetuated the need for techniques, measures and tools for automated knowledge discovery and reasoning about text data. *Terms* are actual words as they occur in texts. *Concepts* are higher level representations of the terms. For example, the term sequence "Madeleine Albright" might represent the higher level concept "Secretary of State". While a concept is a single idea, it can be represented by a single or multiple terms. The relationships between terms and concepts can be expressed in *ontologies*. An ontology specifies the elements that exist in a given domain, and the relationship between these elements. A *link* is two terms or concepts and the relation between them. The relation between two terms or concepts can differ in strength, directionality, and type. A *network* is the union of links.

Content Analysis

A powerful method for analyzing terms and concepts is content analysis: Content analysis is a "research technique for the objective, systematic, and quantitative description of manifest content of communications" that is primarily used in the social sciences and in communication science to analyze media content (Berelson, 1952, p. 18). Performing content analysis involves the following steps:

- Predefining terms to look for in the text data. These terms are noted down in a codebook. Often, specific terms are associated with concepts. The relevant terms can be derived from theory, from data samples, or be provided by subject matter experts.
- Counting the occurrence of terms per unit of analysis, where the unit of analysis can be, for instance, a news article, a web page, or a dialogue.
- Comparing frequency counts across units of analysis.

Network Analysis of Texts

While content analysis considers frequency distributions of the occurrence and co-occurrence of terms and concepts, it does not take the relationships between these terms into account. Networks of words fill this gap. Text data and network data are fundamentally different: Network data consist of nodes and edges, and additional information that help to interpret the nodes and edges (Alderson, 2008). In contrast to that, text data are said to be *sequential*. This is because the basic units of language are naturally arranged in a linear fashion: sentences follow one by one, and the same is true for words and for letters. In general, relational text analysis methods aim to encode links among words in texts and construct a relational representation of the linked words. In semantic networks, the terms and concepts that are derived from text data are represented as nodes. Many types of relationships between these nodes are possible. Typically, these relationships are represented as the existence, frequency, probability, or type of a link between nodes. Converting texts into networks allows for reducing rich qualitative data to structured variables. Going from texts to networks has helped people in answering questions such as:

- What mental model of a certain phenomena does an individual or a group have, and how does this model change over time and compare to the models held by others? (K.M. Carley & Palmquist, 1991)

- Who talks with whom about what? (Danowski, 1993)
- How do innovations, trends and memes emerge, spread and vanish in certain ecosystems? (Adar & Adamic, 2005; Milroy & Milroy, 1985)
- What groups promote or suppress what ideas, and how successful are they in that? (Giuffre, 2001)
- Who are the key players in a socio-technical network, what are their tasks, and what resources and knowledge do they have? Which benefits and risks may the observed network structure imply? (K. M. Carley, Diesner, Reminga, & Tsvetovat, 2007)

Information from text data can help to reveal and/ or enhance information about networks. Respective scenarios can be divided into the following two categories:
First, information from text data might complement given network data. One example for this scenario are network questionnaires that ask the respondents not only to indicate relationships between people, but also to answer some open-ended question that inquire further information about individuals, their relations, or even the entire network. Answers to such open-ended questions are typically provided in text form. Another example are blogs, where relationships between people and the network as a whole are not only expressed through explicit pointers, but also through the blog entries and comments that people write, share, and use as means of engaging in dialogue.
Second, texts are sometimes the only source for information about a network. Most instances of this situation can be categorized as one or more of the following cases:

- Networks that do not exist anymore, such as extinct cultures and bankrupt companies.
- Very large networks. Here, the collection of network data via surveys within the network boundaries is prohibitively expensive. Examples include anthropological studies of sizable ethnic groups.
- Covert networks, such as white collar crime coalitions and adversarial group.
- Networks that are not manifested in a real-world social networks, such as online networks or networks that equal the data that are generated by or within them, such as blogs.

These data may include information about socio-technical networks and people's cognition, both of which can sometimes be represented as relational data and be further analyzed with network analytical methods. Furthermore, today's communication theories are oriented towards complex, large-scale systems. Thus, analysts need powerful tools and methods in order to gain multi-level access to the meaning of textual data (Corman, et al., 2002; P. R. Monge & N. Contractor, 2003). From a network analytical point of view, effective and efficient methods are needed that support the extraction of relevant, user-defined instances of node and edge classes from unstructured, natural language text data. This chapter introduces you such a method. The network text analysis methods discussed in this book follow the workflow model displayed in Figure 1.

Figure 73: Workflow for going from texts to networks

Network Text Analysis in AutoMap

Text analysis in AutoMap is a computer-supported process. Some routines are fully automated, while others require decisions from the user. In general, the user can make choice about the following steps that are part of text coding and that impact the properties and quality of the results

1. Text pre-processing: a collection of routines that help to clean the data and reduce it to the terms and concepts that are relevant with respect to the user's questions. Pre-processing supports users in making meaningful interpretations and comparisons across texts (Ryan & Bernard, 2000).
2. Node identification: select an appropriate strategy for determining the terms and concepts that are implicitly or explicitly represented in the texts data and will be converted into nodes.
3. Edge identification: select an appropriate strategy for linking nodes into edges.

All three steps together form a *coding scheme*. In the next three chapters, we describe the coding choices offered in AutoMap in detail, and their impact on analysis results (for a more detailed description of the impact of coding see (Diesner & Carley, 2004).
AutoMap can extract two types of networks:

1. Semantic Networks: these are one-mode networks. Generating semantic networks requires text pre-processing and node and edge location.
2. Meta Networks: these are multi-mode networks. Generating meta networks requires text pre-processing, node and edge location, and node classification. Here, classification means to assign to type to nodes.

Brining the categories described in this chapter together results in the options and requirements specified in Table 1.

Table 4: Methods covered in book

	Pre-Processing	**Node and edge location**	**Node classification**

Semantic Networks	yes		
Meta Networks	yes		
Covered in Chapter	X	X+1	X+2

In Action

This chapter describes how to:

- Load data into AutoMap
- Extraction relevant, non-relation information from texts
- How to reduce the data down to the terms that will be converted into nodes

There are no scientific standards for determining the best pre-processing strategy for a given research question, data set, and content domain. This chapter introduces you to the most common pre-processing techniques, describes the mechanism, advantages, limitations, and typical use cases.

Input

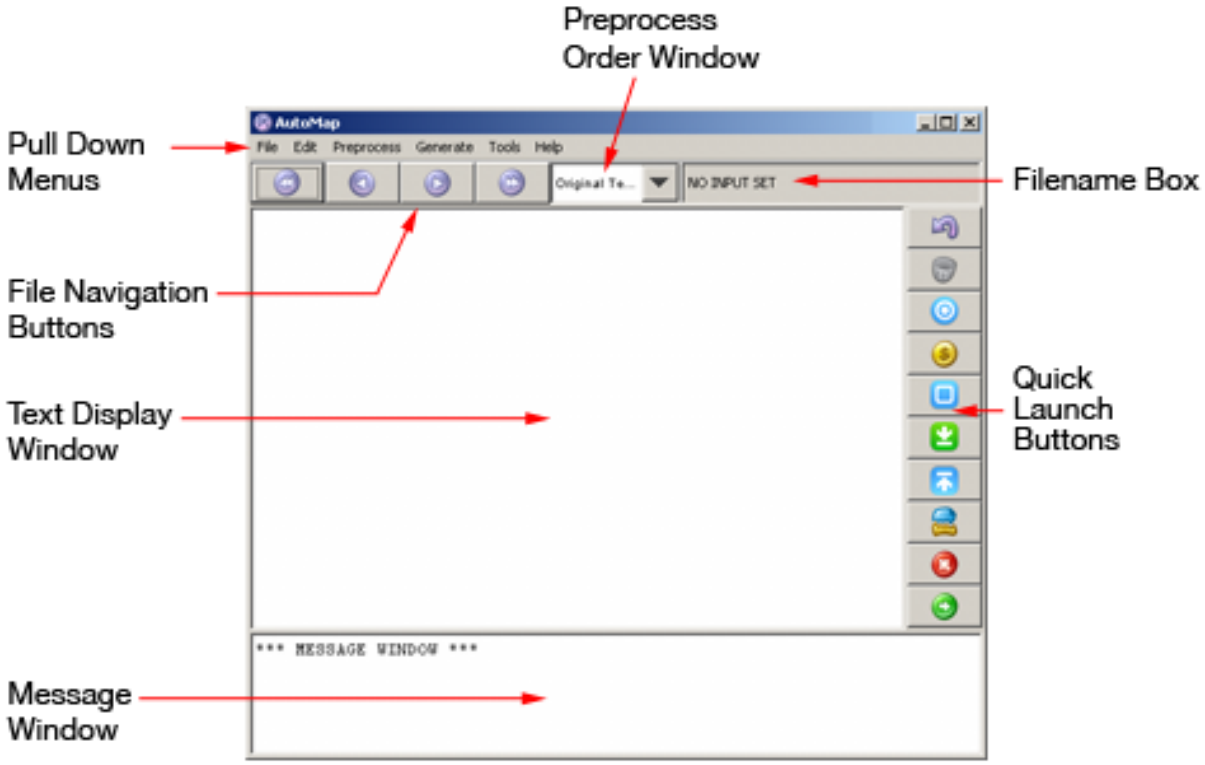
AutoMap accepts any file in txt format as input. If your text data is stored in different formats, or can either convert it into txt or use AutoMap's import routines.

Text Preprocessing

AutoMap Interface

The GUI (Graphical User Interface) contains access to AutoMap's features via the menu items and shortcut buttons.

Table 5: AutoMap Main GUI



The following contains a description of all these elements.

- The Pull Down Menu contains the following main elements:
  - File: Used for loading and saving text files.
  - Edit: Allows the user to change the font of the Display Window
  - Preprocess: Where all the preprocessing of files is done before generating any output. These functions alter original text files only.
  - Generate: Used for the generation of output from preprocessed files. These functions output files based on work done with preprocessing tools.
  - Tools: AutoMap contains a number of Editors and Viewers for the files. These allow the user to view support files used in preprocessing.
  - Help: The Help file and about AutoMap.
- File Navigation Buttons: Used to display the files in the main window. The buttons contain from left to right: First, Previous, Next, and Last
- Preprocess Order Window: Contains a running list of the preprocesses performed on the files. This can be undone one process at a tine with the Undo command.
- Filename Box: Displays the name of the currently active file. Using the File Navigation Buttons will change this and as well as the text displayed in the window.

- Text Display Window: Display the text for the file currently listed in the Filename Box.
- Message Window: Area where AutoMap display the actions taken as well errors encountered.

Import Text

First, select an input directory. In order to do that, place all your text files in an empty directory and use Select Input Directory to load them into AutoMap. All files in the directory will be loaded.
AutoMap asks for the type of text encoding to use. Let AutoMap Detect AutoMap will attempt to import text with the best possible encoding method. A character encoding system consists of a code that pairs a sequence of characters from a given character set (sometimes incorrectly referred to as code page) with something else, such as a sequence of natural numbers, octets or electrical pulses, in order to facilitate the transmission of data (generally numbers and/or text) through telecommunication networks and/or storage of text in computers. If you are not sure which encoding to use, chose the "Let AutoMap detect" option.
The following encodings are supported:
Western : A standard character encoding of the Latin alphabet. It is less formally referred to as Latin-1. It was originally developed by the ISO, but later jointly maintained by the ISO and the IEC. The standard, when supplemented with additional character assignments (in the C0 and C1 ranges: 0x00 to 0x1F and 0x7F, and 0x80 to 0x9F), is the basis of two widely-used character maps known as ISO-8859-1 (note the extra hyphen) and Windows-1252.
UTF-16 : (Unicode Transformation Format) is a variable-length character encoding for Unicode, capable of encoding the entire Unicode repertoire. The encoding form maps each character to a sequence of 16-bit words. Characters are known as code points and the 16-bit words are known as code units. For characters in the Basic Multilingual Plane (BMP) the resulting encoding is a single 16-bit word. For characters in the other planes, the encoding will result in a pair of 16-bit words, together called a surrogate pair. All possible code points from U+0000 through U+10FFFF, except for the surrogate code points U+D800–U+DFFF (which are not characters), are uniquely mapped by UTF-16 regardless of the code point's current or future character assignment or use.
GB2312 : The registered internet name for a key official character set of the People's Republic of China, used for simplified Chinese characters. GB abbreviates Guojia Biaozhun (????), which means national standard in Chinese.
Big5 : The original Big5 character set is sorted first by usage frequency, second by stroke count, lastly by Kangxi radical. The original Big5 character set lacked many commonly used characters. To solve this problem, each vendor developed its own extension. The ETen extension became part of the current Big5 standard through popularity.

Save Preprocessed Text Files

Saves all text files at the highest level of preprocessing. This procedure can be done any number of times during processing. Just make sure if you want to keep a set of files to save them to an empty directory.

Undo

When you want to go back a step in processing, se the "Undo" option. This routine Removes the last Preprocessing done to the text. Does only one step at a time. Multiple Undos can be performed on the text.

Concept Lists

A Concept List is all the concepts of one individual file. Using a Concept List a text can be refined using other functions such as a Delete List (to remove unnecessary concepts) and Generalization Thesaurus (to combine n-grams into single concepts). The number of unique concepts considers each concept only once, whereas the number of total concepts considers repetitions of concepts.

Union Concept Lists

The Union Concept List differs from the Concept List in that it considers concepts across all texts currently loaded, rather than only the currently selected text file. The Union Concept List is helpful in finding frequently occurring concepts, and after review, can be determined as concepts that can be added to the Delete List.

Deletion

A Delete List is a list of concepts to be removed from a text files. It is primarily used to reduce the number unnecessary concepts. By reducing the number of concepts being processed run times are decreased and semantic networks are easier to understand. This also helps in the creation of a semantic network in reducing the number of superficial nodes in ORA.
The Delete List is NOT case sensitive. He and he are considered the same concept. Placing either one in the Delete List will move all instances. You can create Delete Lists from a text editor or use the tools in AutoMap to assist in creating a specially-tailored Delete List. All Delete Lists can be edited. Multiple Delete Lists can be used on the same set of files. Any Delete List can be saved and used for any other text files.
For the most part using a Delete List on a file is a good idea. It removes many concepts that are unnecessary as they do not affect the meaning of the major concepts. But in some style of documents the meaning of two bi-grams could be drastically affected by two seemingly useless words. Most Delete Lists contain the concepts the and a. These two definite articles usually do not change the meaning of the text. But in some instances the meaning could be very substantial. In a Field Operations manual there is a definite difference between the terms a response and the response. It is subtle, but very important. So before you use a Delete List make sure that the words being included are not going to change the meaning.

Remove Extra White Space

Removes all cases of multiple white spaces and replaces them with a single space. Find instances of multiple spaces and replaces them a single space. After running removing extra whitespace all instances of multiple spaces are reduced to a single space. The practice of putting two spaces at the end of a sentence is a carryover from the days of typewriters with mono-spaced typefaces. Two spaces, it was believed, made it easier to see where one sentence ended and the next began. Most typeset text, both before and after the typewriter, used a single space. Today, with the prevalence of proportionally spaced fonts, some believe that the practice is no longer necessary and even detrimental to the appearance of text.

Remove Punctuation

The Remove Punctuation function removes the following punctuation from the text: .,:;' "()!?-. The option is to remove completely or replace with a white space.

Remove Symbols

The list of symbols that are removed: ~`@#$%^&*_+=[]\|/<>. The option is to remove completely or replace with a white space.

Remove Numbers

Removing numbers will remove not only numbers as individual concepts but also removes numbers embedded within concepts. The option is to remove completely or replace with a white space.

Stemming

Stemming is a process for removing the commoner morphological and inflectional endings from words in English. It detects inflections and derivations of concepts in order to convert each concept into the related morpheme. This assists in counting similar concepts in the singular and plural forms (e.g. plane and planes would normally be considered two terms). After stemming planes becomes plane and the two concepts are counted together. This can be broken down into two subclasses, Inflectional and Derivational. Inflectional morphology describes predictable changes a word undergoes as a result of syntax (the plural and possessive form for nouns, and the past tense and progressive form for verbs are the most common in English). These changes have no effect on a word's part-of-speech (a noun still remains a noun after pluralizations). Derivational morphology may or may not affect a word's meaning (e.g.; '-ise', '-ship'). Although English is a relatively weak morphological language, languages such as Hungarian and Hebrew have stronger morphology where thousands of variants may exist for a given word. In such a case the retrieval performance of an IR system would be severely be impacted by a failure to deal with such variations.

The Krovetz Stemmer

K-STEM or Krovetz stemmer effectively and accurately removes inflectional suffixes in three steps, the conversion of a plural to its single form (e.g. '-ies', '-es', '-s'), the conversion of past to present tense (e.g. '-ed'), and the removal of '-ing'. The conversion process firstly removes the suffix, and then though a process of checking in a dictionary for any recoding (also being aware of exceptions to the normal recoding rules), returns the stem to a word. This Stemmer is frequently used in conjunction with other Stemmers, making use of the advantage of the accuracy of removal of suffixes by this Stemmer.

Porter Stemming

The Porter stemmer uses the Porter Stemming algorithm. Additionally, it converts irregular verbs into the verb's infinitive. Each language works it's stems differently. It's important to use the correct language files when stemming else you will obtain incorrect results.
There is a difference in the way the Porter and K-Stem functions stem words: consider(s) and dairy. Porter removes both the er and the ers from the words consider and considers. K-Stem removes the s from considers and both words end up as consider. Porter changes the y/span> in dairy to an i whereas K-Stem leaves the word untouched.
In AutoMap, you can decide whether or not to stem capitalized words. This will include all proper nouns. NOTE : If capitalized words are not stemmed then remember that the first word of each sentence will likewise not be stemmed.

BiGrams

BiGrams are two adjacent concepts in the same sentence (two concepts can not cross sentence or paragraph boundary). If a Delete List is run previous to detecting bi-grams then the concepts in the Delete List are ignored. Multiple Delete Lists can be used with a set of files. Threshold is used to detect if there are specific number of occurrences of a Bi-Gram in the text(s). For Global Threshold a Bi-gram is detected if the total number of its occurrences in all texts is >= Global Threshold. For Local Threshold a Bi-gram is detected if the number of its occurrences in EACH text is >= Local Threshold.

Key Term Extraction

The Feature Selection creates a list of concepts as a TF*IDF (Term Frequency by Inverse Document Frequency) descending order. This list can be used to determine the mot important concepts in a file.

Thesauri

Generalization means applying a thesaurus. A thesaurus has two columns. The first column lists text terms to look for. The second column specifies the concepts that the text terms are to be specified with. Creating an appropriate and exhaustive requires significant expertise.
In AutoMap, a thesaurus contains the following elements:

1. Every line contains a term that is comma separate from the concept to replace the term with. The syntax is text_term, replacement_term.
2. The text_term can be one or more words in a row.s
3. A Key concept must be one word.
4. The Thesaurus is not case sensitive.

There are multiple uses for thesauri:

1. Combining multi-word concepts: Peoples names usually consist of two or more individual names like John Smith or Jane Doe. John Smith becomes John_Smith. It is also useful if, after the initial presentation of the full name, a person is referred to by only part of that name. The thesauri would be able to create one concept out of either entry. John Smith becomes John_Smith John becomes John_Smith.
2. Normalizing abbreviations: Many large companies and organizations are recognized by the abbreviation of their name as well as the name itself. The British Broadcasting Company is routinely known as the BBC. The Chief Executive Officer of a company is known as the CEO. Be aware that some ordinary words can be misinterpreted as organizations. One notable example is WHO - World Health Organization.
3. Normalizing contraction: Contractions are used to shorten two concepts into one smaller concept. isn't => is not | I'd => I would | they'll => they will Expanding these contractions out to their roots allows for creating better Delete Lists.
4. Correcting typos: When typing people routinely make small spelling errors. Many of these are done when people are not sure of the correct spelling. absense,absence | centruy,century | manuever,maneuver Or correcting common typing mistakes hte instead of the or chaor instead of chair.

The order of the entries in the thesauri is important. If an entry toward the beginning contains part of an entry that follows it then both substitutions will be done. This will result in an incorrect thesauri replacement. In the following example carter is substituted first causing incorrect substitutions later on.
Thesauri can be applied in two different ways:

1. Using Thesaurus content only (thesauri as positive filters): After all concepts are replaced by key concepts from the thesaurus then only concepts matching those from the thesaurus will be kept.
2. When using this option, another decision needs to be made about adjacency:
    a. Direct adjacency: means that original distances between concepts that represent the key concepts are neither visualized nor considered for analysis.

   b. Rhetorical adjacency: means that the original distances between key concepts are retained and incorporated into later analyses. The original distances are visually symbolized by placeholders (xxx).
3. Not using Thesaurus content only (thesauri used for normalization): Any concepts not in the thesaurus remain unaffected. All concepts, whether they are contained in the thesaurus or not, are output.

Parts of Speech

Parts of Speech tagging assigns a single best Part of Speech (POS), such as noun, verb, or preposition, to every word in a text. While many words can be unambiguously associated with one tag, (e.g. computer with noun), other words can match multiple tags, depending on the context that they appear in.
AutoMap offers two POS taggers (Table 3): one that outputs the Penn tree Bank POS, and one that outs an aggregated POS set. The PTB divides verbs into six subgroups (base form verbs, present participle or gerund verbs, present tense not 3rd person singular verbs, present tense 3rd person singular verbs, past participle verbs, past tense verbs). In some applications you might want to aggregate these into one verb group. Also, for certain purposes, the union of all prepositions, conjunctions, determiners, possessive pronouns, particles, adverbs, and interjections could be collected into one group that represents irrelevant terms.
AutoMap's POS tagger is based on a Hidden Markov Model training process (Diesner & Carley, 2008). A Hidden Markov model (HMM) is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters; the challenge is to determine the hidden parameters from the observable data. The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications. An HMM can be considered as the simplest dynamic Bayesian network.

Table 6: Parts of Speech available in AutoMap: Penn Trebank (PTB) and aggregated set

PTB Tag	Meaning	Aggregated Tag
NN	noun, common, singular or mass	NOUN
IN	preposition or conjunction, subordinating	IRR
DT	determiner	IRR
JJ	adjective or numeral, ordinal	ADJ
NNP	noun, proper, singular	AGENT
NNS	noun, common, plural	NOUN
RB	adverb	IRR
PRP	pronoun, personal	ANA
VBD	verb, past tense	VERB
CC	conjunction, coordinating	IRR
VB	verb, base form	VERB
VBN	verb, past participle	VERB
TO	to as preposition or infinitive marker	IRR
VBZ	verb, present tense, 3rd person singular	VERB
VBG	verb, present participle or gerund	VERB
PRP$	pronoun, possessive	IRR
CD	numeral, cardinal	NUM
VBP	verb, present tense, not 3rd person singular	VERB
MD	modal auxiliary	MODAL
:	:	SYM
''	''	SYM
``	``	SYM
POS	genitive marker	POS
WDT	WH-determiner	IRR
WP	WH-pronoun	IRR
WRB	Wh-adverb	IRR
JJR	adjective, comparative	ADJ
)	)	SYM
(	(	SYM
EX	existential there	IRR
NNPS	noun, proper, plural	ORG
RBR	adverb, comparative	IRR
JJS	adjective, superlative	ADJ
RP	particle	IRR
SYM	symbol	SYM
UH	interjection	IRR
FW	foreign word	FW
RBS	adverb, superlative	IRR
PDT	pre-determiner	IRR
$	$	SYM
LS	list item marker	SYM
WP$	WH-pronoun, possessive	IRR

These parts of speech can be used in several ways:

Table 4: Usage of aggregated tags

Aggregated Tag	Meaning	Possible Usage
IRR	Irrelevant term	Deletion
NOUN	Noun	resources
VERB	Verb	Events
ADJ	Adjective	Attributes
AGENT	Agent	Social Entity Extraction
ANA	Anaphora	anaphora resolution
SYM	Noise	Deletion
NUM	Number	Attributes
MODAL	Modal verb	Edge types
POS	Genetive marker	no use yet
ORG	Organization	Social Entity Extraction
FW	Foreign Word	no use yet

Problem sets

For the exercises in this section, use the XXX dataset and the ORA and AutoMap software.

1. Generate a union concept list.
   1.1. What are the top 10 terms?
   1.2. What percentage of terms appears only once?
   1.3. Plot the cumulative frequency of words on the y axis and the words on the x-axis. What distribution do you observe?

2. Based on the union concept list, construct a delete list.
   2.1. Apply the delete list with rhetorical adjacency.
   2.2. Regenerate the union concept list.
   2.3. What are the top 10 terms?
   2.4. What percentage of terms appears only once?
   2.5. Plot the cumulative frequency of words on the y axis and the words on the x-axis. How did the distribution change?

3. Identify the salient concepts based on:
   1.1. Frequency after deletion, but no other pre-processing
   1.2. Frequency after deletion and stemming
   1.3. Tf*dif
   1.4. Compare the resulting lists with respect to:
      1.4.1.Top 10 terms
      1.4.2.Length of list
   1.5. Rank the lists according to your judgment about the most relevant top-scoring terms.

4. Run tf*idf on the entire corpus in the following ways:
   4.1. without any other pre-processing
   4.2. after deletion
   4.3. after deletion and K-Stem
   4.4. Compare the top 10 terms per list. What are commonalities, what are differences?

5. Generate a bigram list. Use the bigram list, tf*idf, and the most salient terms as identified above to construct a small generalization thesaurus. Apply the generalization thesaurus to the data in the following ways:
   5.1. Positive filter, no delete list application.
   5.2. Normalization after delete list application.
   Regenerate a union concept list.
   5.3. For each of the two cases, what are the top 10 terms?

Further readings

A large body of methods and theories for the relational representation of language and information has been developed across multiple disciplines. Examples include:

- Linguistics, e.g. discourse representation theory (Kamp, 1981).
- Cognition Science, e.g. spreading activation (Collins & Quillian, 1969).
- Social Sciences, e.g. map analysis (K.M. Carley & Palmquist, 1991) and structural models built via qualitative text coding and grounded theory (Glaser & Strauss, 1967).
- Political science, e.g. event coding (King & Lowe, 2003; Schrodt, Yilmaz, Gerner, & Hermick, 2008).
- Communication Science, e.g. Centering Resonance Analysis (Corman, et al., 2002) and Latent Semantic Indexing (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990).
- Computing, e.g. machine learning approaches (Pearl, 1988), and the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001).

These methods exhibit various commonalities and differences. However, the underlying assumption with many of these approaches is that relations provide context that contributes to the meaning of text data and thereof constructed network data (Mohr, 1998; Sowa, 1992).

CHAPTER 13: Extracting Networks from Texts – Semantic Networks

Extraction of semantic networks

Introduction

Networks originating from text data are referred to as semantic networks. Semantic networks are knowledge representation schemes involving nodes and links between nodes (J. Sowa, 1992). It is a way of representing relationships between concepts. The nodes represent concepts and the links represent relations between nodes. The links are directed and labeled; thus, a semantic network is a directed graph. Semantic networks can help to answer the following questions:

- How uses what words?
- How do people connect or associate the words they use?
- What words to people use in order to refer to more abstract themes?
- What themes do people evoke?
- How do people link the themes they evoke?

Objective

The objective of this chapter is to teach you how to perform the extraction of one-mode networks form texts with AutoMap, and how to appropriately visualize analyze the resulting networks in ORA.

Background

Semantic networks are theoretically grounded in the assumption is that language and knowledge can be modeled as networks of words and the relations between them (J. F. Sowa, 1984; Woods, 1975).
The links between nodes can be identified based on proximal (Danowski, 1993; Doerfel, 1998), logical (Shapiro, 1971; Woods, 1975), syntactic (Gerner, Schrodt, Francisco, & Weddle, 1994; Janas & Schwind, 1979) and semantic information (K.M. Carley, 1994; K.M. Carley & Palmquist, 1991; Van Atteveldt, 2008).
Semantic networks have been used for many purposes. One example are mental models, i.e. structured representations of certain phenomena that people have in their minds and use to make sense of their surroundings. Such cognitive constructs are assumed to reflect the subjects' knowledge and information about a certain topic (K.M. Carley, 1988; Klimoski & Mohammed, 1994; Rouse & Morris, 1986).
AutoMap uses a distance-based approach called Windowing (Danowski, 1993). Changing the window size determines the span which connections will be made. The larger the window size, the more connections within that window. The window slides allow the text changing by one concept as each window is finished. Starting at the beginning it will analyze the set of concepts. The window will then move one concept to the right and create a new window to analyze. This will continue until it reaches the end of the text. Determining a correct window size is important. Too small and important links may be missed. Too large and too many concepts are connected and important links may be overwhelmed.
The outputs that AutoMap generates reflect the complexity of the original input texts and the author's mental model. Therefore, semantic network analysis as performed in AutoMap allows the user to stay close to the data, but to also represent data in a rich network structure that provides contextual information.

In action

Extracting semantic networks requires choices about node and edge identification.

Node Identification Strategies

Using thesauri as positive filters as explained in the previous chapter is an efficient and common strategy for the reducing the text to the most relevant concepts in a computer-assisted fashion. Semantic network extraction can be run on original input texts, or after deletion and/or generalization.

Edge Identification Choices

In AutoMap, the window can be customized on several dimensions:

- Select Directionality : Directionality can be either:
  - unidirectional, i.e. making connections between two concepts by only looking forward in the text limited by the window size, or
  - bidirectional, i.e. making connections between two concepts by looking both forward backward in the text limited by the window size.
- Select Window Size : The distant concepts can be and still have a relationship to one another. Only concepts in same window can form statements. The window is defined in textUnit.
- Select Stop Unit : The limiting factor for concepts to make a connection. The text unit can be comprised of one of the following:
  - Sentence: a sentence is a grammatical unit of one or more words.
  - Word: A word is a unit of language that represents a concept which can be expressively communicated with meaning
  - Clause: A clause consists of a subject and a verb. There are two types of clauses: independent and subordinate (dependent). An independent clause consists of a subject verb and also demonstrates a complete thought: for example, "I am sad". A subordinate clause consists of a subject and a verb, but demonstrates an incomplete thought: for example, "Because I had to move".
  - Paragraph: A paragraph is indicated by the start of a new line. It consists of a unifying main point, thought, or idea accompanied by supporting details. All : The entire text
- Select Number of Sentences: The number of sentences that can be included with the Window Size parameter.

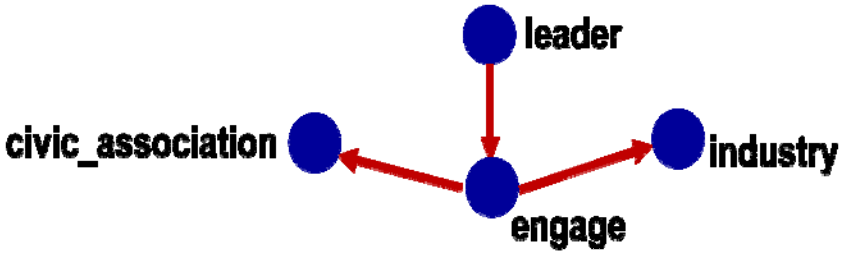
To give an example for the window approach as implemented in AutoMap, given the following thesaurus, the following combination of distance based features results in the network shown in Figure 1:
Thesaurus:

- Leader, leader
- Involved, engage
- Industry, industry
- civic associations, civic_association

Coding choices:

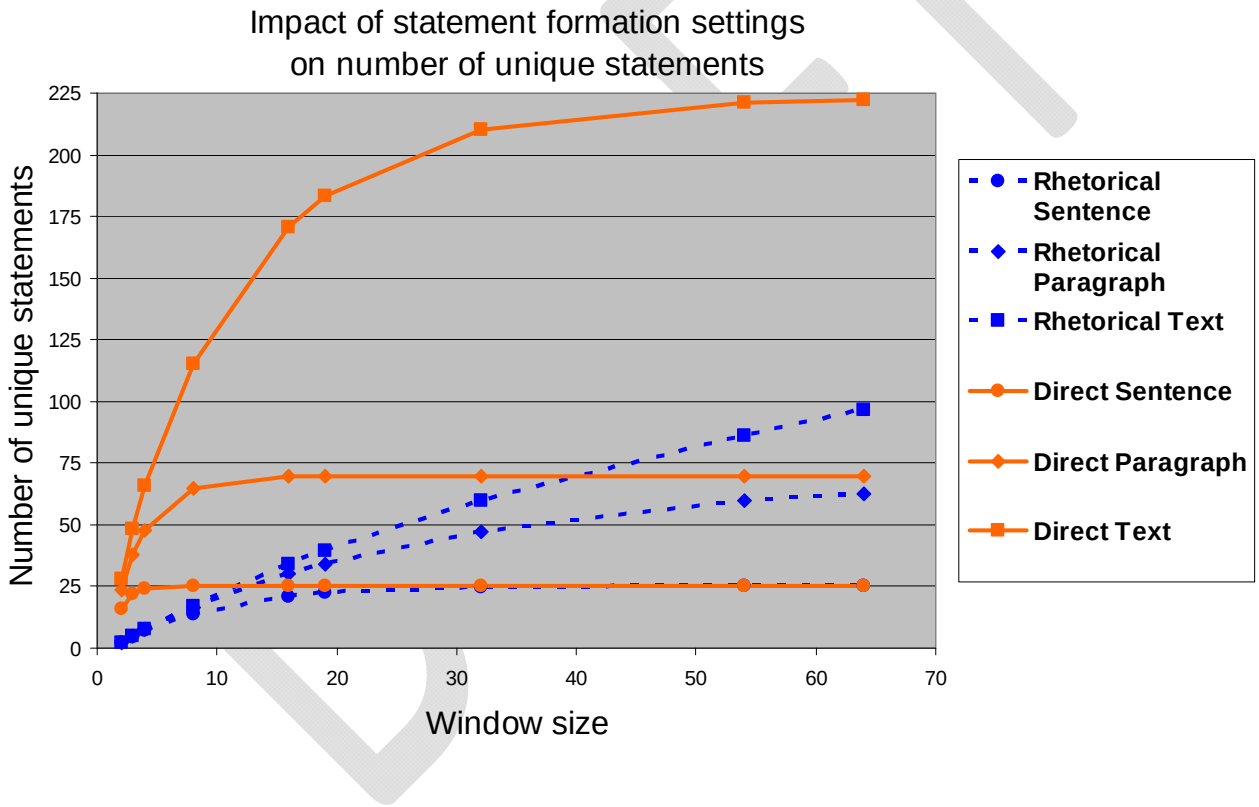
- Stop unit:Sentence
- Thesaurus: content only, rhetorical adjacency
- window size: 5

Figure 74: Resulting network



Different link formations choices lead to different networks extracted from texts. In a previous study (Lewis, Carley, & Diesner, 2003) we measured the impact of various text units, window sizes, and adjacency options on the respective networks (see Figure 2).

Figure 2: Relationship between coding choices and number of statements



Analyses in ORA

Once the data have been extracted, they can be loaded into ORA. ORA is a network analysis tool that detects risks or vulnerabilities of an organization's design structure. ORA contains over 100 measures which are categorized by which type of risk they detect. ORA generates formatted reports viewable on screen or in log files, and reads and writes networks in multiple data formats to be interoperable with existing network analysis packages. In addition, it has tools for graphically visualizing network data and for optimizing a network's design structure.

Two of the reports in ORA that are specifically designed for analyzing semantic networks, or networks of words in general, are the semantic networks report for comparing networks, and the hot topics report. The reports and respective measures are described in more detail in (Kathleen M. Carley, Reminga, Storrick, & DeReno, 2009).

Hot topics report

The hot topics report creates will reveal the most frequent concepts and the least frequent concepts. Given a semantic network, this report displays statistics on the concepts and links in the map, and the distribution of concept frequencies. This report takes unstructured information into account. This is similar to performing content analysis. The user can vary the number of highest scoring terms to output.

Semantics networks report

The semantics networks report analyzes one or more semantic networks, computes the central graphs, and key concepts and links. This is a comparison of two or more semantic networks. Each node in the network is considered a concept, and each link connects two concepts. Link weights are interpreted as the number of times it occurred in the underlying input text.

The report includes tests whether the two semantic networks are statistically different using four different T-tests, based on the number of nodes, the number of links, betweenness centrality, and degree centrality. The symmetric distance of network A to network B is the number of concepts and links in A that are not in B. In set theoretic terms, this is the set difference (A - B).

This report also provides communicative power analysis. Concepts are classified according to whether they have high and low values for the measures: total degree centrality, betweenness centrality, and consensus. The concept classes are based on (K.M. Carley, 1997):

- Ordinary (low degree, low betweenness, low consensus)
- Factoids (low degree, low betweenness, high consensus)
- Buzzwords (low degree, high betweenness, low consensus)
- Emblems (low degree, high betweenness, high consensus)
- Allusions (high degree, low betweenness, low consensus)
- Stereotypes (high degree, low betweenness, high consensus)
- Placeholders (high degree, high betweenness, low consensus)
- Symbols (high degree, high betweenness, high consensus)

The report also creates new networks that represent different levels of agreement between the loaded networks. The set-theoretic intersection on the 25, 50, 75% percent level is considered for this comparison. More precisely, the following outputs are computed:

- The union of compared links.
- The consensus between networks as it is represented by the intersection of links.
- The dissension between networks as it is represented as the difference.

The report outputs the nodes and edges in the intersection, offset, and union. Performing map comparison can help to answer questions such as:

- Do different people use the same words and themes in the same way?
- Do different people link concepts and themes in the same way?
- Do different people share the same knowledge?
- How similar or different are the analyzed texts?

Problem sets

6. Generate two semantic networks based on the operations performed for the problem set in the previous chapter, task 4.1:
   6.1. Establish links across the entire text by using a window size of seven and rhetorical adjacency. Do not perform parts of speech tagging for that part.
   6.2. Establish links across the entire text by using a window size of seven and direct adjacency. Do not perform parts of speech tagging for that part.

7. Load both semantic networks into ORA .
   7.1. Visualize the networks.
      7.1.1. Which terms appear key?

8. In ORA, run the hot topics report.
      8.1.1. Which terms are key in both networks?
      8.1.2. Compare both networks based on the following metrics and interpret your results:
         8.1.2.1. Number of nodes
         8.1.2.2. Number of links
         8.1.2.3. Density

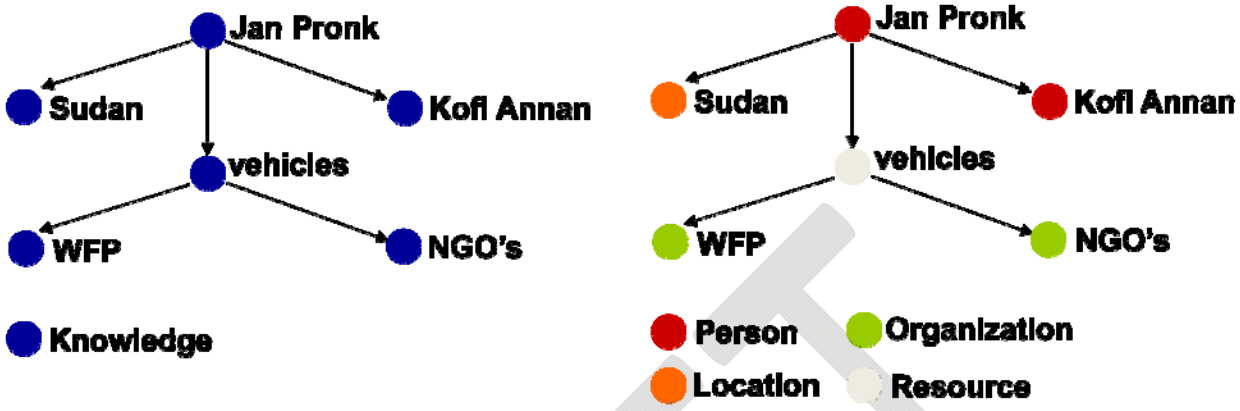
9. In ORA, run the semantic networks report.
      9.1.1. Which nodes appear in both networks?
      9.1.2. What percentage of nodes is shared by both networks?
      9.1.3. Which links appear in both networks?
      9.1.4. What percentage of links is shared by both networks?

CHAPTER 14: Extracting Networks from Texts – Meta-Networks

Extraction of meta networks

The semantic networks extracted in the previous chapter had nodes that were all of the same type. Multi-mode networks allow you for moving beyond this approach by classifying nodes into different categories. In AutoMap, the meta-matrix approach is used as a baseline ontology for node types (Carley, 2002; Diesner & Carley, 2005; Krackhardt & Carley, 1998). The meta network approach is a representational framework and a set of methods for analyzing multimode data. The original model aimed to combine theoretically grounded insight from of knowledge management, operations research, organization science, social network analytic techniques and measures (Krackhardt & Carley, 1998).
Figure 1 provides an illustrative example for the difference between semantic networks and meta networks.

Example from UN News Service (New York), 12-28-2004:
"Jan Pronk, the Special Representative of Secretary-General Kofi Annan to Sudan, today called for the immediate return of the vehicles to World Food Programme (WFP) and NGOs."



Objective

The objective of this chapter is to teach you to perform the extraction of multi-mode networks form texts, and to appropriately analyze these data in ORA by using various reports. A secondary learning goal is the introduction of the distinction between one-mode networks, such as semantic networks, and multimode networks, such as socio-technical networks that involve agents and their infrastructures.

Background

Today, Network Analysis refers to the analysis of any network such that all the nodes are of one type (e.g., all people, or all roles, or all organizations), or at most two types (e.g., people and the groups they belong to). The metrics and tools in this area, since they are based on the mathematics of graph theory, are applicable regardless of the type of nodes in the network or the reason for the connections. For most researchers, the nodes are actors. As such, a network can be a cell of terrorists, employees of global company or simply a group of friends. However, nodes are not limited to actors. A series of computers that interact with each other or a group of interconnected libraries can comprise a network also. Figure one shows the types of nodes considered in the meta network model.

Figure 75: meta network model

Meta-Matrix	Agent	Knowledge	Resource	Task/ Event	Organization	Location
Agent	Social network	Knowledge nw	Capabilities nw	Assignment nw	Membership nw	Agent location nw
Knowledge		Information nw	Training nw	Knowledge requirement nw	Org. know-ledge nw	Knowledge location nw

Resource			**Resource nw**	**Resource requirement nw**	**Org. capabilities nw**	**Resource location nw**
Task/ Events				**Precedence nw**	**Org. assign-ment nw**	**Task/Event location nw**
Organization					**Inter-Org. nw**	**Org. location nw**
Location						**Proximity nw**

Meta network analysis examines the connections between and among these entity classes. A clsssical social network, for example, means connects between nodes of the type *agent*. Moving beyond the social network analysis approach allows you to ask more questions about socio-technical networks, such as:

- Who are the key players in a socio-technical system?
- Where are they located, what are their tasks, and what resources or knowledge do they have?
- Which benefits or risks does the observed network structure imply?

In Action

In AutoMap, nodes of the following classes can be identified and extracted:

- Who (agent, organization)
- What (task, event)
- When (date)
- Where (location)
- How (resources, knowledge)
- Why (beliefs)

ORA can reason about all the last of these categories. The next section specifies these categories in more details:

- Agent: A person, group, organization, or artificial actor that has information processing capabilities. All whos are agents whether they be a person in a group, a group within an organization, or the organization itself (e.g. President Barack Obama, the shadowy figure seen outside the building, or the Census bureau).
- Organization: A group of agents working together for a common cause (e.g. The Red Cross or the local chess club). It is up to the user's discretion what sub-category to place these agents in.
- Task : A task is part of a set of actions which accomplish a job, problem or assignment. Task is a synonym for activity although the latter carries a connotation of being possibly longer duration
- Knowledge: Information learned such as a school lecture or knowledge learned from experience (e.g. Excellent knowledge of the periodic table or "I know what you did last summer").

- Resource: Can be either a physical or intangible object. Anything that can be used for the completion of a job. (e.g. Use a car to drive from point A to point B or use money from a bank account to fund something).
- Event: Something that happens, especially something of importance. Events are usually thought of as a public occasions but they can also be clandestine meetings. The number of agents can range in the thousands or as few as two agents (e.g. Christmas in Times Square or dinner with friends).
- Location: An actual physical place. This could be a room in a building, a city, or a country (e.g. Pittsburgh, PA or my living room).
- Role: An agents role can be defined as their job for their employer or the part they serve during an event.
- Attribute: Information about the specifics of the agents. These are usually traits that agents have in common, each can be slightly different (e.g. visible traits like hair colour or intangible traits like religious beliefs).
- When: Referring to time or circumstances. Can be as broad as a year or as pinpoint as the exact time of a particular day (e.g. Last year or 2:33 PM on March 1st, 2009).

In AutoMap, meta network extraction can be run on texts that were pre-processed with a meta-matrix thesaurus. Similar to the thesauri described in the previous chapter, a meta-matrix thesaurus associates concepts with meta-network categories. When applying a meta-network thesaurus, AutoMap associates the words specified in the left hand side column with the categories specified in the right hand side column. A concept can be translated into one, multiple or none meta-network category. All pre-processing steps described in the previous chapter can be applied prior to this procedure. Meta network are always applied with thte "thesaurus content only" option in AutoMap. At this point, the user needs to make a decision about the adjacency option as explained din the previous chapter. Meta network thesauri can be built by hand or automatically.

Manual meta network thesaurus construction

Associating the right hand side entries of a generalization thesaurus as the left hand side entries for the meta matrix thesaurus in a useful starting point. Additionally, the union concept list can be screened for further concepts to be considered in the meta network thesaurus. One concept might need to be translated into several meta network categories. For example, "Carnegie Mellon" might represent an *agent*, a *location*, or a *resource*.

Sometimes, external sources are very helpful in constructing meta network thesauri. For example, the CIA world fact book (Central Intelligence Agency) provides lists of the parties and party leaders for each country.

Automated meta network thesaurus construction

AutoMap can provide an initial suggestion for a meta network thesaurus that is subject to further user verification. The thesaurus is provided based on a machine learning approach that is based on a probabilistic graphical models (Diesner & Carley, 2008). This model has an approximate prediction accuracy of 83.6% for locating and classifying nodes.

Problem sets

1. Manually build a meta network thesaurus for the XXX dataset. Spend not more than 20 minutes on this exercise. Use all meta network categories, i.e. find at least one term for each category.
   1.1. How many different terms did you consider?
   1.2. How many terms did you consider per category? Rank the categories by decreasing frequency of assigned terms.

2. Use the automated meta network thesaurus construction technique in order to construct a meta network thesaurus.
   2.1. Compare this thesaurus to the one you constructed in task 1. How many terms do the have in common? How many terms are in the manually built file, but not in the automatically constructed one, and vice versa?
   2.2. Screen the thesaurus for false positives and false negatives. Correct the thesaurus accordingly.

3. Apply the automatically constructed meta network thesaurus to the xxx data. Load the generated network into ORA.
   3.1. Visualize the network.
   3.2. Run the key entities report in ORA.
   3.3. Run the standard social networks analysis in ORA.

References

Adar, E., & Adamic, L. (2005). Tracking Information Epidemics in Blogspace. Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, Compiegne, France, September 2005.

Alderson, D. (2008). Catching the'network science'bug: Insight and opportunity for the operations researcher. Operations Research, 56, 1047-1065.

Barnlund, D. and C. Harland (1963). "Propinquity and Prestige as Determinants of Communication Networks." Sociometry 26(4): 467-479.

Berelson, B. (1952). Content analysis in communication research. Glencoe, Ill: Free Press.

Bernard, H., & Ryan, G. (1998). Text analysis: Qualitative and quantitative methods. In H. Bernard (Ed.), Handbook of methods in cultural anthropology (pp. 595–646). Walnut Creek: Altamire press.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. Scientific American, 284(5), 34-43.

Butts, C. and K. Carley (2000). Spatial Models of Large-Scale Interpersonal Networks. Pittsburgh, PA, Carnegie Mellon University: 1-47.

Carley, K. (1991). "A Theory of Group Stability." American Sociology Review 56(3): 331-354.

Carley, K. (1995). "Communication Technologies and their Effect on Cultural Homogeneity, Consensus, and the Diffusion of New Ideas." Sociological Perspectives 38(4): 547-571.

Carley, K. (1999). "On the Evolution of Social and Organizational Networks." Special Issue of Research in the Sociology of Organizations on Networks In and Around Organizations: 3-30.

Carley, K. M. (1988). Formalizing the Social Expert's Knowledge. Sociological Methods & Research, 17(2), 165-232.

Carley, K. M. (1990). Content Analysis. In R. E. Asher (Ed.), The Encyclopedia of Language and Linguistics (Vol. 2, pp. 725-730). Edinburgh: Pergamon Press.

Carley, K. M. (1994). Extracting culture through textual analysis. Poetics, 22, 291-312.

Carley, K. M. (1997). Network text analysis: The network position of concepts. In C. W. Roberts (Ed.), Text Analysis for the Social Sciences: Methods for Drawing Statistical Inferences from Texts and Transcripts (pp. 79–100). Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc.

Carley, K. M. (1997). Network text analysis: The network position of concepts. In C. W. Roberts (Ed.), Text Analysis for the Social Sciences: Methods for Drawing Statistical Inferences from Texts and Transcripts (pp. 79–100). Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc.

Carley, K. M. (2002). Smart agents and organizations of the future. In L. Lievrouw & S. Livingstone (Eds.), The Handbook of New Media: Social Shaping and Consequences of ICTs (pp. 206–220). Thousand Oaks, CA: Sage.

Carley, K. M., & Palmquist, M. (1991). Extracting, Representing, and Analyzing Mental Models. Social Forces, 70(3), 601 - 636.

Carley, K. M., & Palmquist, M. (1991). Extracting, Representing, and Analyzing Mental Models. Social Forces, 70(3), 601 - 636.

Carley, K. M., Diesner, J., Reminga, J., & Tsvetovat, M. (2007). Toward an interoperable dynamic network analysis toolkit. Decision Support Systems. Special Issue Cyberinfrastructure for Homeland Security, 43(4), 1324-1347.

Carley, K. M., Reminga, J., Storrick, J., & DeReno, M. (2009). ORA User's Guide 2009 (No. Technical Report CMU-ISR-09-115): Carnegie Mellon University, School of Computer Science, Institute for Software Research.

Carley, K., M. Martin, et al. (2009). "The Etiology of Social Change." Topics in Cognitive Science 1(3).

Central Intelligence Agency World Factbook: Available from: https://www.cia.gov/library/publications/the-world-factbook/.

Collins, A., & Quillian, M. (1969). Retrieval Time from Semantic Memory. Journal of Verbal Learning & Verbal Behavior, 8(2), 240-248.

Corman, S. R., Kuhn, T., McPhee, R. D., & Dooley, K. J. (2002). Studying Complex Discursive Systems: Centering Resonance Analysis of Communication. Human Communication Research, 28(2), 157-206.

Danowski, J. A. (1993). Network Analysis of Message Content. Progress in Communication Sciences, 12, 198-221.

Danowski, J. A. (1993). Network Analysis of Message Content. Progress in Communication Sciences, 12, 198-221.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American society for information science, 41(6), 391-407.

Diesner, J., & Carley, K. M. (2004). AutoMap1.2 - Extract, analyze, represent, and compare mental models from texts (No. CMU-ISRI-04-100): Carnegie Mellon University, School of Computer Science, Institute for Software Research International.

Diesner, J., & Carley, K. M. (2005). Revealing Social Structure from Texts: Meta-Matrix Text Analysis as a novel method for Network Text Analysis. In V. K. Narayanan & D. J. Armstrong (Eds.), Causal Mapping for Information Systems and Technology Research: Approaches, Advances, and Illustrations (pp. 81-108). Harrisburg, PA: Idea Group Publishing.

Diesner, J., & Carley, K. M. (2008). Conditional Random Fields for Entity Extraction and Ontological Text Coding. Journal of Computational and Mathematical Organization Theory, 14, 248 - 262.

Diesner, J., & Carley, K. M. (2008). Looking under the hood of machine learning algorithms for parts of speech tagging (No. CMU-ISR-08-131R): Carnegie Mellon University, School of Computer Science, Institute for Software Research.

Doerfel, M. (1998). What Constitutes Semantic Network Analysis? A Comparison of Research and Methodologies. Connections, 21(2), 16-26.

Epstein, J. and R. Axtell (1999). Growing Artificial Societies, MIT Press.

Gerner, D., Schrodt, P., Francisco, R., & Weddle, J. (1994). Machine Coding of Event Data Using Regional and International Sources. International Studies Quarterly, 38(1), 91-119.

Giuffre, K. (2001). Mental Maps: Social Networks and the Language of Critical Reviews. Sociological Inquiry, 71(3), 381-393.

Gladwell, M. (2000). The Tipping point: how little things can make a big difference, Little Brown Inc %@ 0-316-31696-2.

Glaser, B., & Strauss, A. (1967). The Discovery of Grounded Theory: Strategies for Qualitative Research. New York, NY: Aldine.

Harrison, D., K. Price, et al. (1998). "Beyond Relational Demography: time and the effects of surface- and deep-level diversity on group cohesion." Academy of Management Journal 41(1): 96-107.

Hirshman, B. and J. St. Charles (2009). Simulating Emergent Multi-Tierd Social Ties. Proceedings of the 2009 Human Behavior and Computational Intelligence Modeling Conference. Oak Ridge National Laboratory, TN.

Hirshman, B. and K. Carley (2007). Specifying Agents in Construct, Carnegie Mellon University School of Computer Science.

Hirshman, B. and K. Carley (2007). Specifying Networks in Construct, Carnegie Mellon University School of Computer Science.

Hirshman, B. and K. Carley (2008). Modeling Information Access in Construct, Carnegie Mellon University School of Computer Science.

Hirshman, B., K. Carley, et al. (2009). Decisions, Variables, and Scripting in Construct. Carnegie Mellon University School of Computer Science, Carnegie Mellon University.

Hirshman, B., M. Martin, et al. (2008). The Impact of Educational Interventions by Socio-Demographic Attribute, Carnegie Mellon University School of Computer Science.

Janas, J., & Schwind, C. (1979). Extensional Semantic Networks. In N. V. Findler (Ed.), Associative Networks. Representation and Use of Knowledge by Computers. (pp. 267 - 302). New York, San Francisco, London: Academic Press.

Kamp, H. (1981). A Theory of Truth and Semantic Representation Formal Methods in the Study of Language. In J. A. G. Groenendijk, T. M. V. Janssen & M. B. J. Stokhof (Eds.), Formal Methods in the Study of Language. (pp. 277-322): Mathematical Centre Tracts 135, Amsterdam.

Kapferer, B. (1972). Strategy and Transaction in an African Factory: African workers and Indian management in a Zambian town. Manchester, Manchester University Press.

King, G., & Lowe, W. (2003). An Automated Information Extraction Tool for International Conflict Data with Performance as Good as Human Coders: A Rare Events Evaluation Design. International Organization, 57(03), 617-642.

Klimoski, R., & Mohammed, S. (1994). Team Mental Model: Construct or Metaphor? Journal of Management, 20(2), 403.

Krackhardt, D., & Carley, K. M. (1998). A PCANS Model of Structure in Organization. Paper presented at the International Symposium on Command and Control Research and Technology, Monterray, CA, June.

Lewis, E. T., Carley, K. M., & Diesner, J. (2003). Concept Networks In Organizational Language: Consensus or Creativity?, XXIII Sunbelt Social Network Conference. Cancun, Mexico.

McPherson, M., L. Smith-Lovin, et al. (2001). "Birds of a Feather: Homophily in Social Networks." Annual Review of Sociology 27: 415-444.

Milroy, J., & Milroy, L. (1985). Linguistic change, social network and speaker innovation. Journal of Linguistics, 21, 339-384.

Mohr, J. (1998). Measuring Meaning Structures. Annual Reviews in Sociology, 24(1), 345-370.

Monge, P. R., & Contractor, N. (2003). Theories of Communication Networks. New York: Oxford University Press.

Monge, P., & Contractor, N. (2003). Theories of Communication Networks: Oxford University Press, USA.

Moon, I.-C. and K. Carley (2007). Estimating the Near-Term Changes of Organization with Simulation. AAMAS, Honolulu, HI.

Pearl, J. (1988). Probabilistic reasoning in intelligent systems: networks of plausible inference. San Francisco, CA: Morgan Kaufmann.

Rouse, W. B., & Morris, N. M. (1986). On looking into the black box; prospects and limits in the search for mental models. Psychological Bulletin(100), 349-363.

Ryan, G., & Bernard, H. (2000). Data management and analysis methods. In N. Denzin & Y. Lincoln (Eds.), Handbook of qualitative research (2nd ed., pp. 769-802): Sage.

Schrodt, P. A., Yilmaz, Ö., Gerner, D. J., & Hermick, D. (2008). Coding Sub-State Actors using the CAMEO (Conflict and Mediation Event Observations) Actor Coding Framework. Paper presented at the Annual Meeting of the International Studies Association, San Francisco, CA, March 2008.

Shapiro, S. (1971). A net structure for semantic information storage, deduction and retrieval. Proceedings of the Second International Joint Conference on Artificial Intelligence,

Simon, H. (1957). A Behavioral Model of Rational Choice. Models of Man: Mathematical Essays on Rational Human Behavior in a Social Setting. London, England, John Wiley & Sons, ltd: 241-260 %@ 57-5933.

Sowa, J. (1992). Semantic Networks. In S. C. Shapiro (Ed.), Encyclopedia of Artificial Intelligence (2nd ed., pp. 1493 - 1511). New York, NY, USA: Wiley and Sons.

Sowa, J. (1992). Semantic Networks. In S. C. Shapiro (Ed.), Encyclopedia of Artificial Intelligence (2nd ed., pp. 1493 - 1511). New York, NY, USA: Wiley and Sons.

Sowa, J. F. (1984). Conceptual Structures: Information Processing in Mind and Machine. Reading, MA: Addison-Wesley.

Steglich, C., T. Snijders, et al. (2006). "Applying Sienna: an illustrative analysis of the co-evolution of adolescents' friendship networks, taste in music, and alcohol consumption." Methodology 2(1): 48-56.

Van Atteveldt, W. (2008). Semantic network analysis: Techniques for extracting, representing, and querying media content. Charleston, SC: BookSurge Publishers.

van Cuilenburg, J., Kleinnijenhuis, J., & de Ridder, J. (1986). A Theory of Evaluative Discourse: Towards a Graph Theory of Journalistic Texts. European Journal of Communication, 1(1), 65.

Wegner, D. (1986). Transactive memory: A Contemporary Analysis of the Group Mind. Theories of group behavior, New York, Springer-Verlag.

Woods, W. (1975). What's in a link: Foundations for semantic networks. In D. Bobrow & A. Collins (Eds.), Representation and Understanding: Studies in Cognitive Science (pp. 35-82). New York, NY: Academic Press.

Zhou, W. X., D. Sornette, et al. (2005). "Discrete Hierarchical Organization of Social Group Sizes." Proceedings of the Royal Society of Biological Sciences 272: 439-444.

Glossary

***A* |**

Adjacency Matrix – A Matrix that is a square actor-by-actor (i=j) matrix where the presence of pair wise edges are recorded as elements. The main diagonal, or self-tie of an adjacency matrix is often ignored in network analysis.

Aggregation – Combining statistics from different entities to higher entities.

Algorithm – A finite list of well-defined instructions for accomplishing some task that, given an initial state, will terminate in a defined end-state.

Attribute – Indicates the presence, absence, or strength of a particular connection between entities in a Matrix.

***B* |**

Betweenness – Degree an individual lies between other individuals in the network; the extent to which an entity is directly connected only to those other entities that are not directly connected to each other; an intermediary; liaisons; bridges. Therefore, it's the number of entities who an entity is connected to indirectly through their direct links.

Betweenness Centrality – High in betweenness but not degree centrality. This entity connects disconnected groups, like a Go-between.

Bimodal Network – A network most commonly arising as a mixture of two different unimodal networks*.

Binarize – Divides your data into two sets; zero or one.

Bipartite Graph – Also called a bigraph. It's a set of entities decomposed into two disjoint sets such that no two entities within the same set are adjacent.

***C* |**

Centrality – The nearness of an entity to all other entities in a network. It displays the ability to access information through edges connecting other entities. The closeness is the inverse of the sum of the shortest distances between each entity and every other entity in the network.

Centralization – Indicates the distribution of connections in the employee communication network as the degree to which communication and/or information flow is centralized around a single agent or small group.

Classic SNA density – The number of edges divided by the number of possible edges not including self-reference. For a square matrix, this algorithm* first converts the diagonal to 0, thereby ignoring self-reference (an entity connecting to itself) and then calculates the density. When there are N entities, the denominator is (N*(N-1)). To consider the self-referential information use general density.

Clique – A sub-structure that is defined as a set of entities where every entity is connected to every other entity.

Clique Count – The number of distinct cliques to which each entity belongs.

Closeness – Entity that is closest to all other Entities and has rapid access to all information.

Clustering coefficient – Used to determine whether or not a graph is a small-world network.

Cognitive Demand – Measures the total amount of effort expended by each agent to do its tasks.

Column Degree – see Out Degree.

Complexity – Complexity reflects cohesiveness in the organization by comparing existing links to all possible links in all four networks (employee, task, knowledge and resource).

Complementarity – The idea that people seek others with characteristics that are different from and complement their own, aka the idea that opposites attract.

Concor Grouping – Concor recursively splits partitions and the user selects n splits. (n splits -> 2^n groups). At each split it divides the entities based on maximum correlation in outgoing connections. Helps find groups with similar roles in networks, even if dispersed.

Congruence – The match between a particular organizational design and the organization's ability to carry out a task.

Construct – A reduced form of Construct is found in ORA in the Near Term Impact Report.

Count – The total of any part of a MetaMatrix row, column, entity, edge, isolate, etc.

CSV – File structure meaning Comma Separated Value. Common output structure used in database programs for formatting data.

***D* |**

Degree – The total number of edges to other entities in the network.

Degree Centrality – Entity with the most connections. (i.e. In the know). Identifying the sources for intel helps in reducing information flow.

Density –

- **Binary Network:** The proportion of all possible edges actually present in the Matrix.
- **Value Network:** The sum of the edges divided by the number of possible edges. (i.e. the ratio of the total edge strength that is actually present to the total number of possible edges).

Dyad – Two entities and the connection between them.

Dyadic Analysis – Statistical analysis where the data is in the form of ordered pairs or dyads. The dyads in such an analysis may or may not be for a network.

Dynamic Network Analysis – Dynamic Network Analysis (DNA) is an emergent scientific field that brings together traditional Social Network Analysis* (SNA), Link Analysis* (LA) and multi-agent systems (MAS).

DyNetML – DynetML is an xml based interchange language for relational data including nodes, ties, and the attributes of nodes and ties. DyNetML is a universal data interchange format to enable exchange of rich social network data and improve compatibility of analysis and visualization tools.

***E* |**

Edge – A specific relation among two entities. Other terms also used are tie and link.

Entities – General things within an entity class (e.g. a set of actors such as employees). Can be who, what, where, how, why, a thing that is being studied.

For example, entities can be ***people, agents, organizations, beliefs, expertise areas, resources, tasks, events, locations***.

Entity – A specific entity (e.g., Joe, Martha, Bob; or, airplanes, buses, bicycles).

Entity Class – The type of items we care about (knowledge, tasks, resources, agents) or a set of entities of one type.

Entity Level Metric – is one that is defined for, and gives a value for, each entity in a network. If there are x entities in a network, then the metric is calculated x times, once each for each entity. Examples are Degree Centrality*, Betweenness*, and Cognitive Demand*.

Entity Set – A collection of entities that group together for some reason.

Eigenvector Centrality – Entity most connected to other highly connected entities. Assists in identifying those who can mobilize others

***F* |**

FOG – (F)uzzy (O)verlapping (G)roups. Gives a better understanding of individuals spanning groups. Fuzzy groups are a more natural and compelling way of thinking of human social groups.

***G* |**

General density – The number of edges divided by the number of possible edges including self-reference. For a square matrix, this algorithm includes self-reference (an entity connecting to itself) when it calculates the density. When there are N entities, the denominator is (N*N). To ignore self-referential information use classic SNA Density.

Geodesic Distance – A generalization of the notion of a straight line to curved spaces. In presence of a metric, geodesics are defined to be (locally) the shortest path between points on the space.

Gini coefficient – The measure of inequality of a distribution of income. Uses a ratio with values between 0 and 1: the numerator is the area between the Lorenz curve of the distribution and the uniform (perfect) distribution line; the denominator is the area under the uniform distribution line.

Graph Level Metric – A metric defined for, and gives a value for, the network as a whole. The metric is calculated once for the network. Examples are Centralization, Graph Hierarchy, and the maximum or average Betweenness.

GraphML – GraphML is a comprehensive and easy-to-use file format for graphs. It consists of a language core to describe the structural properties of a graph and a flexible extension mechanism to add application-specific data.

Group - A collection of things, such as entities, nodes, ties, and networks. A group might at times be represented as a meta-node.

Nodes may be classified in groups based on a shared attribute, type, id-range, label, user selection, etc.

For example: if you have a set of people and know their gender, then there might be two groups - men and women.

In addition, the nodes representing those people could be displayed as a meta-node for men and a meta-node for women.

Nodes may be classified into groups based on a grouping algorithm. For example, if you have a network showing connections among members of an organization and you run a grouping algorithm it will return clusters of nodes that fit together on some mathematical criteria. This cluster is a group and can be represented as a meta-node.

***H* |**

Homophily – (i.e., love of the same) is the tendency of individuals to associate and bond with similar others.

- Status homophily means that individuals with similar social status characteristics are more likely to associate with each other than by chance.
- Value homophily refers to a tendency to associate with others who think in similar ways, regardless of differences in status.

***I* |**

In-Degree – The sum of the connections leading to an entity from other entities. Sometimes refered to row degree.

Influence network – A network of hypotheses regarding task performance, event happening and related efforts.

Isolate – any entity which has no connections to any other entity

***L* |**

Label - Each entity, node, link or network has a type - what kind it is, is called a label. This can be the common name, an id - unique identifying number or some other kind of label.

Lattice Network – A graph in which the edges are placed at the integer coordinate points of the n-dimensional Euclidean space and each entity connects to entities which are exactly one unit away from it.

Link – The representation of the *tie, connection, relation, edge between two nodes*.

Link Class - A set of links of one type. A set of links of one type can be represented as a meta-link as well.

Link Analysis – A scientific area focused on the study of patterns emerging from dyadic observations. The relationships are typically a form of co-presence between two entities. Also multiple dyads that may or may not form a network.

***M* |**

Main Diagonal – in a sqaure matrix this is the conjunction of the rows and cells for the same entity.

Math Terms

These mathematical terms and symbols are used: Let S be any set:

- card(S) = |S| = the cardinality of S (the cardinality of the entity-sets is represented as |A|, |K|, |R|, |T|)
- $\Re$ denotes a real number
- Z denotes an integer

Matrix Algebra – The part of algebra that deals with the theory of matrices.

Measure – A measure is a function that maps one or more networks to R^n. Measures are often scalar (n=1) or vector valued with n=|V| or n=|U|.

MetaMatrix – A statistical graph of correlating factors of personnel, knowledge, resources and tasks. These measures are based on work in social networks, operations research, organization theory, knowledge management, and task management.

Meta-Link - The representation of a group of links.

Meta-Network - The representation of a Group of networks.

Meta-Node - The representation of a group of nodes as one node. For example, if Karen, Bill, Tom, and Ed are all accountants, they could all belong to the Meta-Node "Accountants" which would contain all their individual nodes.

Monte Carlo – A random optimization of your organization

Multi-entity – More than one type of entity (people, events, locations, etc.).

Multi-plex – Network where the links are from two or more relation classes.

Multimode Network – Where the entities are in two or more entity classes.

***N* |**

Neighbors – Entities that share an immediate edge to the entity selected.

Network – Set of links among entities. Entities may be drawn from one or more entity classes and links may be of one or more relation classes. The representation of a set of nodes, including meta-nodes, of one type and the links, including meta-links, of one type between them.

Network Class – A set of networks of one type. Note: a set of networks of one type can be represented as a meta-network.

A network N is a triple consisting of two sets of entities, called U and V, and a set of edges $E \subset UxV$. Thus, we write N=(U,V,E). An element e = (i,j) in E indicates a relationship or tie between entities $i \in U$ and $j \in V$. A network where U=V and therefore $E \subset VxV$ is called unimodal otherwise the network is bimodal. We write G=(V,E) for unimodal networks. For our purposes, unimodal networks will not contain self loops, which means that $(i,i) \notin E$ for $i \in V$.

Newman Grouping – Finds unusually dense clusters, even in large networks.

Node – The representation of a single entity (a who, what, where, how why item). This pertains to actually looking at a network model. In the visual model, entities are called nodes.

Node Class - A set of nodes of one type. Note: a set of nodes of one type can be represented as a meta-node.

Notation:

The following matrix notation is used throughout the document for an arbitrary matrix X:

- X(i,j) = the entry in the i^{th} row and j^{th} column of X
- X(i,:) = i^{th} row vector of X
- X(:,j) = j^{th} column vector of X
- sum(X) = sum of the elements in X (also, X can be a row or column vector of a matrix)
- dich(X) = dichotomize (make binary) X, so that dich(X)(i,j) = 1 iff X(i,j) > 0
- X' = the transpose of X
- ~X = for binary X, ~X(i,j) = 1 iff X(i,j) = 0
- X@X = element-wise multiplication of two matrices (e.g. Z=X@Y => Z(i,j) = X(i,j)*Y(i,j))

O |

ODBC – (O)pen (D)ata (B)ase (C)onnectivity is an access method developed by the SQL Access group in 1992 whose goal was to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data.

Ontology – "The Specifics of a Concept". The group of entities, resources, knowledge, and tasks that exist in the same domain and are connected to one another. It's a simplified way of viewing the information.

Organization – A collection of networks.

Out-Degree – The sum of the connections leading out from an entity to other entities. This is a measure of how influential the entity may be. Sometimes refered to as column degree.

P |

Path - A set of nodes and links that form a single unbroken chain, such that no nodes and links can be repeated.

Pendant – Any entity which is only connected by one edge. They appear to dangle off the main group.

Q |

QAP Correlation – Calculates measures of nominal, ordinal, and interval association between the relations in two matrices, and uses quadratic assignment procedures to develop standard errors to test for the significance of association.

R |

Random Graph – One tries to prove the existence of graphs with certain properties by assigning random edges to various entities. The existence of a property on a random graph can be translated to the existence of the property on almost all graphs using the famous Szemerédi regularity lemma*.

Reciprocity – The percentage of entities in a graph that are bi-directional.

Redundancy – Number of entities that access to the same resources, are assigned the same task, or know the same knowledge. Redundancy occurs only when more than one agent fits the condition.

Relation – The way in which entities in one class relate to entities in another class.

Robustness – Two different definitions:

- Networks – Concerned with the reliability (Kim & Médard, 2004) and continued functioning of a network following an intervention. The robustness of a network is particularly relevant in communication-type and flow-oriented networks. The purpose for understanding robustness of a network has more of a management of the network connotation.
- Measures – This meaning has more of a statistical connotation. Studying the robustness of a measure of a network can also be referred to as conducting a sensitivity analysis on the measure. In keeping with the terminology of the most-recently published research in this area, in lieu of using the term sensitivity, we too will use the robustness term, although the terms can be used interchangeably.
  A measure is robust if a slight perturbation in its input produces a slight change in its output.

Row Degree – see In Degree*.

***S* |**

Scale-Free Network – Some entities act as highly connected hubs (high degree), although most entities are of low degree. Scale-free networks' structure and dynamics are independent of the system's size N, the number of entities the system has. A network that is scale-free will have the same properties no matter what the number of its entities is.

Self-Loop – An entity with a connection to itself.

Simmelian Ties – Two entities are Simmelian Tied to one another if they are reciprocally and strongly tied to each other and strongly tied to at least one third party in common.

Simulated Annealing – A method of finding optimal values numerically. It's a search method as opposed to a gradient based algorithm. It chooses a new point, and (for optimization) all uphill points are accepted while some downhill points are accepted depending on a probabilistic criteria.

The term Simulated Annealing draws its inspiration from metallurgy, where atoms within a metal are heated thereby dislodging them from a metal's internal structure transforming the metal into another atomic state. In this way, your organization is heated changing its components in the attempt to arrive at an optimized state.

Slow Measures – As the name implies these measures generally take longer to run.

Small-World Network – Small-World Networks will have sub-networks that are characterized by the presence of connections between almost any two entities within them.

Social Network Analysis – The term Social Network Analysis (or SNA) is used to refer to the analysis of any network such that all the entities are of one type (e.g., all people, or all roles, or all organizations), or at most two types (e.g., people and the groups they belong to).

Sphere of Influence – One entity's direct relationship with one of it's neighbors as a function of specified path length.

.stl file format – This file format is native to the stereolithography CAD software created by 3D Systems. STL files describe only the surface geometry of a three dimensional object without any representation of color, texture or other common CAD model attributes and can us both ASCII and binary representations

Szemerédi's Regularity Lemma – A fundamental structural result in extremal graph theory due to Szemerédi (1978). The regularity lemma essentially says that every graph can be well-approximated by the union of a constant number of random-like bipartite graphs, called regular pairs.

T **|**

Tie – see Edge

Topology – The study of the arrangement or mapping of the elements (links, nodes, etc.) of a network, especially the physical (real) and logical (virtual) interconnections between nodes.

Total Degree Centrality – The normalized sum of an entity's row and column degrees.

Trails – Analyzes the trails that an entity class makes through another entity class; for example, how vessels pass through ports. Furthermore, a set of nodes and links that forms a single unbroken chain, such that no node or link is repeated.

Transpose – In linear algebra, the transpose of a matrix A is another matrix A^T (also written A^{tr}, ${}^tA$, or A') created by any one of the following equivalent actions:

- write the rows of A as the columns of A^T
- write the columns of A as the rows of A^T
- reflect A by its main diagonal (which starts from the top left) to obtain A^T

See The Transpose Wikipedia Entry for forumlas, examples and more information.

U **|**

Unimodal networks – These are also called square networks because their adjacency matrix* is square; the diagonal is zero diagonal because there are no self-loops*.

Appendix